Model-Based Chassis Control of a Wheeled Mobile Robot on Soft Ground Using the Example of the ExoMars Planetary Exploration Rover

Vom Fachbereich
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des Grades eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

Stefan Barthelmes, M.Sc.

geboren in Rodez, Frankreich

Referent:	Prof. Dr.-Ing. Ulrich Konigorski
Korreferent:	Prof. Dr.-Ing. Dipl.-Math. Johann Bals
Tag der Einreichung:	21. April 2020
Tag der Prüfung:	13. August 2020

D 17 · Darmstadt 2020

Vorwort

Diese Dissertation habe ich während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Systemdynamik und Regelungstechnik des Deutschen Zentrums für Luft- und Raumfahrt in Kooperation mit dem Fachgebiet Regelungstechnik und Mechatronik der TU Darmstadt erarbeitet.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Ulrich Konigorski für die Betreuung. Unsere fachlichen Diskussionen haben diese Arbeit entscheidend geprägt und seine Fähigkeiten, durch gezielte Fragen innerhalb kürzester Zeit tief ins Thema einzusteigen und mit wertvollen Hinweisen zur Lösung auftretender Probleme beizutragen, haben mich sehr beeindruckt. Herrn Prof. Dr.-Ing. Johann Bals danke ich für die Anstellung am Institut, den fachlichen Diskurs und die Übernahme des Korreferats. Insbesondere bin ich dankbar für den gewährten Freiraum zur Bearbeitung meines Themas und das dafür entgegengebrachte Vertrauen, was keinesfalls selbstverständlich ist.

Weiterhin bedanke ich mich bei meinen Kolleginnen und Kollegen am Institut für die vielen Gespräche und Tipps, sowie die sehr angenehme Arbeitsatmosphäre und grenzenlose Hilfsbereitschaft. Besonders hervorheben möchte ich dabei meinen Bürokollegen Fabian: Ich weiß nicht, ob ich die Dissertation ohne Dich und unsere unzähligen Whiteboard-Sessions durchgestanden hätte. Die Gespräche mit Andreas auf der Zielgeraden, sowie unser kleiner Schreibwettbewerb haben mir schließlich die nötige Motivation gegeben, die Arbeit fertigzustellen - danke! Auch meinem Freund Ferdinand danke ich für die fruchtbaren Diskussionen über regelungstechnische Probleme, schließlich wurde die eine oder andere Idee auf einer gemeinsamen Skitour geboren. Den Studentinnen und Studenten, die Praktika oder Studienarbeiten bei mir absolviert haben danke ich für ihren Beitrag zu dieser Arbeit und die Bereicherung durch unterschiedliche Perspektiven.

Zu guter Letzt folgt mein Dank an diejenigen, die mir durch ihr Vertrauen in meine Fähigkeiten und ihre Ermutigungen die Kraft gegeben haben, diese Dissertation durchzuziehen. Danke Dir, liebe Nicola, dass du immer da warst, mir zugehört hast und mich aufgebaut hast wenn es mal nicht rund lief. Danke Euch, liebe Eltern, dass ihr mir überhaupt erst das Studium ermöglicht habt und während der Promotion immer darauf vertraut habt, dass ich das schon schaffen werde. Nicht zu vergessen ist schließlich der beste Freundeskreis, den man sich wünschen kann: Danke für all die lustigen Abende, gemeinsamen Projekte und natürlich die vielen Urlaube, die als Ablenkung und zum Kräfte sammeln unverzichtbar waren.

München, September 2020
Stefan Barthelmes

Abstract

The exploration of the universe has greatly advanced by the usage of planetary exploration rovers in the last two decades. Major findings were only possible due to the extended range of scientific instruments onboard rovers. NASA's Mars Exploration Rover Opportunity has only stopped operating last year, after 14 years, and thereby exceeded its planned mission duration by a factor of 56. The locomotion system, consisting of single wheel drives and steering actuators as well as passive kinematics, has thus proven to be robust and reliable.

Despite this success, the problem of traversing soft, sandy terrain has been ubiquitous for the NASA rovers. Opportunity's sibling, Spirit, got irretrievably stuck in a soft sand field after breaking through the thin crust that covered it. Opportunity itself almost encountered the same fate when it had wheel slippage of over 98 % for about one month. During the operations of their still active Curiosity rover, the NASA/JPL team decided to avoid certain sand fields in the future. All mentioned and planned rovers feature an *over-actuation* due to simplicity of construction and redundancy reasons, i.e. they have more actuators than they would need for realizing a desired movement. The resulting *degrees of freedom* are, however, not used for optimizing the locomotion but eliminated through geometric constraints in current missions.

This thesis therefore addresses the question whether a *chassis controller* can be designed for the *control allocation* of this system, such that the degrees of freedom are exploited for improving the rover's locomotion in soft sand. Although this work is derived with the example of the Rosalind Franklin rover of the European ExoMars mission, it is transferable to the field of mobile robots on rough terrain in general.

The approach of this thesis is a comprising model development of the robotic rover system and the wheel-ground interaction for soft sand, called *terramechanics*. A fully model-based chassis controller is then synthesized, which, at the same time, allows for explicit computation of the actuator trajectories and exploitation of the full solution space of the over-actuation. For managing these conflicting goals, the full model is divided into dynamic but determined sub-systems and a linear, static, underdetermined system of equations with several necessary model modifications. The nonlinear control methods *feedback linearization* and *dynamic extension* are used for the dynamic sub-systems, whereas the general solution to the underdetermined system of equations is used for the allocation itself. It consists of the *least-squares* solution with a *pseudo-inverse* of the allocation matrix and the homogeneous solution with a base of its *null-space*. The performance of the designed chassis controller is evaluated in a co-simulation with a high-fidelity model of the rover and the robustness of the approach is investigated by the injection of different disturbances.

Zusammenfassung

Die Erforschung des Weltalls wurde in den letzten zwei Jahrzehnten bedeutend durch den Einsatz planetarer Explorationsrover vorangetrieben. Wichtige Erkenntnisse konnten nur durch die erhöhte Reichweite von wissenschaftlichen Instrumenten auf Rovern erzielt werden. Der Rover Opportunity der NASA hat im vergangenen Jahr seine Arbeit nach über 14 Jahren eingestellt und hat damit seine geplante Lebensdauer um das 56-fache überschritten. Das Lokomotionssystem aus Radantrieben, Lenkaktoren und passiver Kinematik hat sich damit als äußerst robust und zuverlässig erwiesen.

Diese Erfolge können dennoch nicht über die Allgegenwärtigkeit von Problemen der Rover mit weichem Sand hinwegtäuschen. Opportunitys Zwillingsrover Spirit grub eines seiner Räder unwiederbringlich in weichem Sand ein, nachdem es durch eine Kruste gebrochen war. Rover sind aus Gründen der einfachen Konstruktion und Redundanz *überaktuiert*. Das bedeutet, dass sie mehr Aktuatoren haben als sie für die Umsetzung der gewünschten Bewegung bräuchten. Die daraus resultierenden *Freiheitsgrade* werden allerdings in aktuellen Missionen durch geometrische Zwangsbedingungen und Annahmen eingeschränkt und nicht für die Optimierung des Vorankommens genutzt. In dieser Arbeit wird daher die Frage adressiert, inwiefern für die genannte *Allokation* ein *Chassisregler* entworfen werden kann, welcher die vorhandenen Freiheitsgrade für eine Verbesserung des Vorankommens in weichem Sand nutzt. Der Rosalind Franklin Rover aus der europäischen ExoMars Mission wird dabei zur Entwicklung des Ansatzes herangezogen, letzterer ist aber auf mobile Roboter in schwierigem Gelände im Allgemeinen übertragbar.

Um die genannte Fragestellung zu beantworten, wird in dieser Arbeit ein vollständiges Modell des robotischen Systems und des Rad-Sand Kontakts, der sogenannten *Terramechanik*, entwickelt. Daraus wird ein modellbasierter Regler synthetisiert, welcher gleichzeitig eine explizite Berechnung der Aktuatortrajektorien und die Nutzung des gesamten Lösungsraums der Überaktuierung ermöglicht. Um diese, zunächst gegensätzlichen, Ziele miteinander zu vereinbaren, werden einige Veränderungen am Gesamtmodell vorgenommen, sodass dieses in dynamische, aber bestimmte Teilmodelle, sowie ein unterbestimmtes, aber statisches lineares Teilmodell aufgeteilt werden kann. Die nichtlinearen Methoden der *exakten Ein-Ausgangslinearisierung* durch Zustandsrückführung, sowie der *dynamischen Erweiterung* werden für die dynamischen Systeme verwendet, während für das lineare, unterbestimmte Gleichungssystem seine allgemeine Lösung verwendet wird. Diese besteht aus der Lösung mit den *kleinsten Quadraten* mit der *Pseudoinverse* der Allokationsmatrix, sowie aus der homogenen Lösung mit einer Basis des *Nullraums*. Die Leistungsfähigkeit des Chassisreglers wird anhand einer Co-Simulation mit einem genauen Modell des Rovers untersucht und seine Robustheit wird durch das Einbringen verschiedener Störungen bewertet.

Contents

Acronyms

ATC2 All-Terrain Controller 2. 84, 85, 87, 88, 89, 90, 91, 92, 94, 95, 97, 99, 101, 102, 103

BB2 Breadboard 2: A development state of the ESA ExoMars rover. 13, 31, 32, 33, 38, 71, 74, 108

CNES Centre National d'Études Spatiales. 3

CNSA China National Space Administration. 2

CoG Center of Gravity. 33, 34, 36, 37, 60, 102, 105

DEM Discrete Element Method. 39

DH Denavit-Hartenberg: A systematic to setup direct kinematics. XIII, 20, 33, 34, 105

DLR Deutsches Zentrum für Luft- und Raumfahrt. 3, 83, 84

DoF Degrees of Freedom. 5, 6, 8, 9, 11, 12, 13, 19, 22, 29, 74, 75, 76, 79, 94, 105

ESA European Space Agency. XI, 2, 4, 6

FDIR Failure Detection, Isolation and Recovery. 109

FEM Finite Element Method. 39

ICR Instantaneous Center of Rotation. 6, 7, 10, 13, 84, 85, 87, 88, 89, 90, 92, 95, 97, 101, 102, 106

IMU Inertial Measurement Unit. 12, 107

JAXA Japan Aerospace Exploration Agency. 3

JPL Jet Propulsion Lab. 2, 4, 6, 84

MER Mars Exploration Rovers. 3

MIMO Multi-Input Multi-Output. 22, 26, 67

MMX Martian Moon eXploration: A JAXA mission to the Martian moon Phobos. 3, 84

NASA National Aeronautics and Space Administration. 2, 4, 84

RPY Roll-Pitch-Yaw. 18, 36, 37, 55, 56, 57, 69, 70, 71, 79

RST Rover Simulation Toolkit. 83, 84, 85, 97, 99, 102

SCM Soil Contact Model. 6, 39, 44, 53, 55, 95, 98, 99, 100, 101, 103

Sol One Martian day. 3, 4

TerRA Terramechanics for Real-time Application: A model for the wheel-soil interaction. 43, 44, 45, 46, 50, 52, 54

UDP User Datagram Protocol. 84

List of Symbols

General Symbols

$\boldsymbol{e}$	Unit vector
$\underline{\boldsymbol{I}}$	Identitiy matrix of matching size
k	Parameter, i.e. a non-dynamic variable of the respective model
$\mathbb{R}$	The set of real numbers
$\underline{\boldsymbol{M}}/\underline{\boldsymbol{m}}$	Matrix (bold, underlined letters)
$\boldsymbol{v}/\boldsymbol{V}$	Vector (bold)

Frames (coordinate systems)

b	Bogie frame with z being the rotation axis of the bogie
ba	Base frame of a kinematic chain in robotics
c	Contact frame which is at the assumed contact point between the respective wheel and the ground with the same orientation as the rover frame
e	End effector frame of a kinematic chain in robotics
0	Inertial, non-moving frame with z-axis pointing in reverse gravity direction
r	Rover frame, which is in the rover's CoG with x, y, z pointing forward, sideways and upwards, respectively
s	Steering frame which is at the respective steering actuator with the same orientation as the rover frame

Kinematics, Differential Kinematics & Dynamics

a	DH parameter: Distance along x-axis
α	DH parameter: Angle about x-axis
d	DH parameter: Distance along z-axis
ϑ	DH parameter: Angle about z-axis

$\boldsymbol{a}_{\mathrm{g}}$	Gravitational acceleration
Θ	Inertia
$\underline{\boldsymbol{J}}$	Jacobian matrix
$\underline{\boldsymbol{M}}_{\mathrm{r}}$	Mass matrix of the rover
q	Joint variable of a prismatic or revolute joint
${}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}}$	Rotation matrix from 'a' to 'b' for rotating a vector from 'b' to 'a'
${}^{\mathrm{c}}\boldsymbol{r}_{\mathrm{b\|a}}$	Position vector of 'b' from 'a' expressed in frame 'c'
${}^{\mathrm{c}}r_{\mathrm{b\|a},x}$	x-component of the position vector of 'b' from 'a' expressed in frame 'c'
ϕ	Roll angle of the RPY convention
θ	Pitch angle of the RPY convention
ψ	Yaw angle of the RPY convention
$\underline{\boldsymbol{S}}$	Cross product building matrix
${}^{0}\underline{\boldsymbol{T}}_{\mathrm{a}}$	Homogeneous transformation matrix from '0' to 'a' for transforming a vector from 'a' to '0'
ω	Rotational velocity / Rate
v	Translational velocity
r	Wheel radius (without index)
w	Wheel width

Wheel-ground Contact Modeling

$\phi_{\mathrm{S-S}}$	Internal friction angle of soil
ϕ_{G}	Gravity-corrected internal soil friction angle
$\phi_{\mathrm{M-S}}$	Friction angle between metal and soil
d	Sinkage of a wheel, i.e. penetration depth into the undisturbed ground
ζ	Angle of soil contact for the integration over the contact patch
$\boldsymbol{z}_{\mathrm{g}}$	Vector of the soil heights at all six contact points
$z_{\mathrm{g},i}$	Soil height at the ith contact point

Controller

$\underline{\boldsymbol{A}}^{+}$	Pseudo-inverse of the $\underline{\boldsymbol{A}}$-matrix
$\underline{\boldsymbol{A}}^{\perp}$	Null-space base of the $\underline{\boldsymbol{A}}$-matrix
ρ	Degree of freedom of the over-actuated system
$\underline{\boldsymbol{V}}$	Decoupling matrix within the feedback linearization of a MIMO system
ξ	Transformed coordinate of a feedback linearization
γ	State transformation function for feedback linearization
η	Internal, i.e. non-observable, states of a closed-loop system
$\boldsymbol{x}$	Vector of system states
$\boldsymbol{u}$	Vector of system inputs
$\boldsymbol{y}$	Vector of system outputs
$\boldsymbol{f}$	Vector of state functions in a non-linear input-affine state space system
$\underline{\boldsymbol{g}}$	Matrix of input functions in a non-linear input-affine state space system
$\boldsymbol{h}$	Vector of output functions in a non-linear input-affine state space system
a	Allocation subsystem
w	Wheel subsystem
$\tilde{\text{w}}$	Modified wheel subsystem after dynamic extension
$(\cdot)^{*}$	A desired value / trajectory
L	Operator for the lie derivative

1 Introduction

In this first chapter the need for a model-based chassis controller for wheeled mobile robots is expounded at the example of planetary exploration rovers in Section 1.2. Subsequently, Section 1.3 summarizes the state of the art of such chassis control systems in the areas of actual exploration missions, in lab research as well as in the related field of over-actuated road vehicles. The particular objectives and the underlying assumptions of this work are set out in Sections 1.4 and 1.5, respectively and the structure of the thesis is explained in Section 1.6.

Section 1.1 is a short guide that proposes sections and figures to look at for those readers that want to get a better understanding of the topic of this work than by only reading the abstract and conclusion but do not have the time to read the whole thesis.

1.1 Guide for a Quick Overview

Although I have written this thesis with the goal of a concise presentation, I understand that only few readers will have the time to read it in its entirety. Therefore I provide this short section to point the reader with little time to the most important sections in order to understand the idea, peculiarities, realization and verification of this work.

First go to Section 1.2 and read the paragraph about *Problems with soft sand* for understanding the motivation. Continue with the subsequent paragraph about *Rover controls hierarchy and the role of chassis controls* for the definition of chassis control and an explanation about how this can help solving the aforementioned problems. Go to Figure 1.4 for an overview of possible concepts and jump right to the objectives and the pursued approach of this work in Section 1.4. Read the model summary in Section 3.9 with the overview of the model in Figure 3.11 and also have a look at Figure 3.1 for an image of the considered rover and Figure 3.2 for its abstraction. The modification of these sub-models is a main part of this work and for understanding why this needs to be done, read the first part of Section 4.1 until you reach (4.6). If you are not familiar with the concept of control allocation, it is worth reading the one-two pages of Section 2.7 beforehand. Then go to the last paragraph of Section 4.1, where the modified model structure is described with the block diagram in Figure 4.2. You are now prepared to read the controller synthesis summary in Section 4.6 and to recognize how the modified model structure is inverted for the controller in Figure 4.4.

After looking at the summary of the simulative verification in Section 5.4, read the conclusion in Chapter 6 for an emphasis of the most important points.

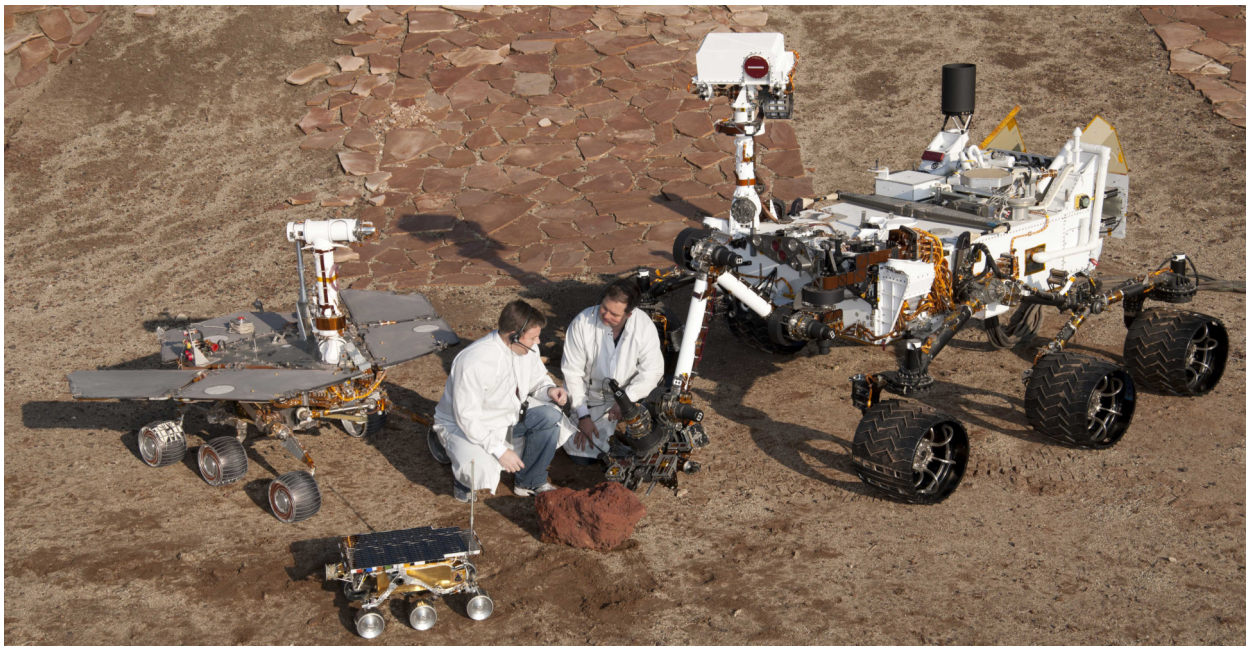


Figure 1.1: Earth siblings of the Sojourner (front), Spirit and Opportunity (left) and Curiosity (right) rovers together with two engineers of the Jet Propulsion Lab (JPL) in their testing area called Mars Yard; courtesy NASA/JPL-Caltech.

1.2 Motivation

The first rover to land on any celestial body was the Russian *Lunokhod 1* in 1970. It was followed by the three austronaut-driven *Apollo Lunar Roving Vehicles* (1971-72) and, shortly after, the *Lunokhod 2* rover (1973) [Ell16]. More than 20 years later, the National Aeronautics and Space Administration (NASA) revived planetary exploration rovers with the Mars Exploration program, see Figure 1.1. The *Sojourner* (1997), *Spirit*, *Opportunity* (both 2004), and *Curiosity* (2011) rovers have been and continue to be vastly successful in exploring Mars. The China National Space Administration (CNSA) focused on the Moon in the last years and landed its first rover *Yutu* in 2014 [Ip+14]. Their second rover, *Yutu-2*, was the first to land on the far side of the Moon in January 2019 [Lin+19].

Numerous rover missions are planned for the near future: NASA scheduled the launch of the *Perseverance* rover, which is of the same size as Curiosity and part of the Mars 2020 mission, in July-August 2020 [Agl19]. The CNSA plans to launch a Mars rover in the same year [LYY18] and seems to be on track according to press releases, e.g. [Jon19]. The European Space Agency (ESA) recently announced to postpone the launch of the *Rosalind Franklin* rover, which is part of the *ExoMars* mission, to 2022 [Eur20]. Within the *Google Lunar X Prize* competition [XPR], numerous privately funded teams have started developing lunar rovers. Although no team was able to meet the deadlines and the competition ended in March 2018, according to news articles, some of the teams announced to continue their efforts [Gru18]. One of them, the Israel-based *SpaceIL*, launched a lander without rover in February 2019 but crashed on the Moon due to a malfunction of the main engine [Fou19]. The

(a) Wheels of Spirit showing high sinkage (especially the wheel on the very left of the image).

(b) Tracks of Curiosity in Hidden Valley.

Figure 1.2: Problems of rovers with soft sand; courtesy NASA/JPL-Caltech.

first rover to drive in the milli-gravity environment of Phobos, one of the Martian moons, is currently under joint development at the French Centre National d'Études Spatiales (CNES) and the German Deutsches Zentrum für Luft- und Raumfahrt (DLR). It is planned to be launched as a payload of Japan Aerospace Exploration Agency (JAXA)'s Martian Moon eXploration (MMX) mission in 2024 [Ber+19].

Why rovers? The main advantage of mobile robots compared to stationary landers is a significantly extended range for the scientific instruments onboard the rover. Manned missions that would provide the same or even higher mobility are much more expensive and dangerous in hostile environments [Ell16]. Wheeled rovers were favored over tracked vehicles, hoppers or walking machines due to their relatively simple construction and robustness. The Mars Exploration Rovers (MER) Spirit and Opportunity are a good example for this robustness. The sibling rovers were both deployed in 2004 with a planned mission duration of 90 Martian days, which are called *Sol* and correspond to roughly 24 Hours and 40 Minutes each [All97]. Despite a failure of one of its six drive actuators on Sol 779 [Arv+10], Spirit was able to continue driving for 1092 Sols. It then continued to work as an immobile instrument until communication was lost in 2010, totaling a driven distance of 7.73 km and a duration of 2210 Sols [Web11]. Opportunity even outlived its sibling by eight years, totaling a driven distance of 45.16 km and an operation duration of 5111 Sols [ABW19], which means a 56-times exceeded planned lifetime.

Problems with soft sand. Despite the success of past and current rover missions, rovers were and are struggling with soft, sandy terrain. The traction force that is needed for any

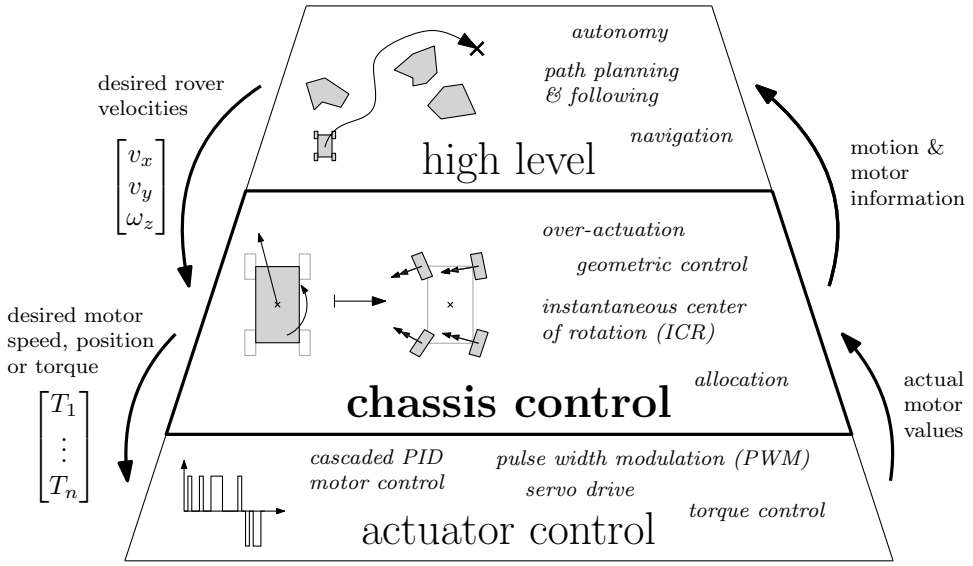


Figure 1.3: Control pyramid for a mobile robot showing the role of a chassis control system. Typical keywords of the respective controller level are written in italic and arrows outside indicate the typical communication interface between them.

movement of the robot is highly dependent on the wheel-ground contact properties and soft sands can drastically decrease the traction potential. Looking closer at the mentioned Mars rover missions, problems with traction on soft sands are very prominent. First and foremost, the Spirit rover got stuck in a sand field named *Sands of Troy* in 2009 after it had buried one wheel into the sand almost up to its diameter, see Figure 1.2a, and a second drive actuator had failed [Arv+10]. Many attempts were made, including a 79 Sol drive pause that was used to replicate the situation on earth and optimize the drive procedures for extrication, until its last movements in February 2010. Opportunity experienced very high slip rates of 98.9 %-99.5 % as well in *Purgatory dune* but was fortunate enough to escape this difficult area after about one month [MCM07]. Even the youngest Mars rover, Curiosity, experienced dangerous slip values in an area called *Hidden Valley*, forcing the operators to back out [Arv+16], see the tracks in Figure 1.2b. In fact, "A decision was made after the Hidden Valley drives to avoid extensive megaripple fields and to drive in valleys covered by regolith, relatively thin windblown sand covers, and single megaripples." [Arv+16].

Rover controls hierarchy and the role of chassis controls. NASA/JPL's last Mars rovers Spirit, Opportunity and Curiosity all have six independently driven wheels, four of which are steerable. A passive chassis kinematic, called *rocker-bogie*, ensures ground contact of all wheels and is meant to distribute normal forces to all six wheels. The ESA Rosalind Franklin rover has six independently driven wheels as well but slightly different 3-bogie kinematics and all six wheels have a steering actuator. The actuation redundancy of all these rovers aims at retaining the drivability even in case of actuator failure and leads to a construction with less moving parts compared to all-wheel drive with single motors requiring transfer gearboxes and drive shafts. For the control of such a rover, a hierarchically

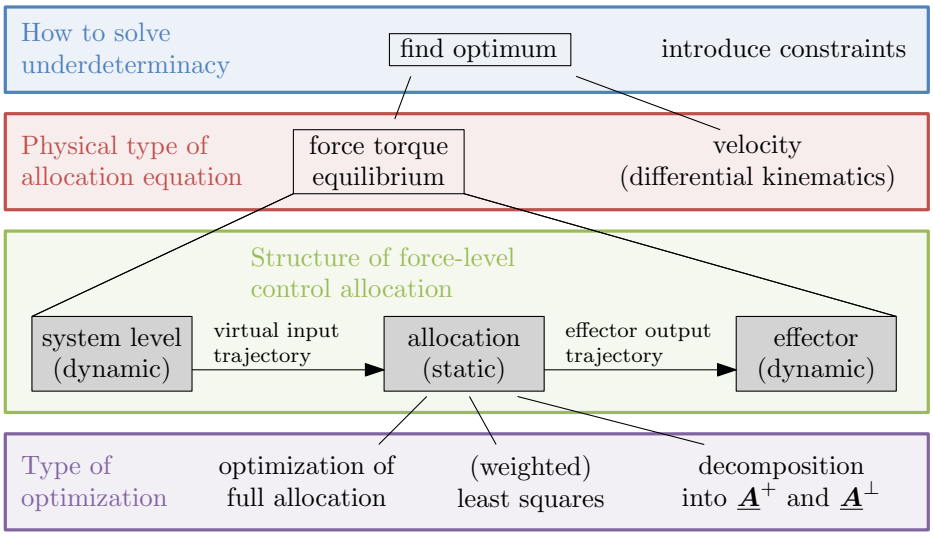


Figure 1.4: Different approaches for the chassis control of an over-actuated wheeled vehicle.

structured control system is often used, see Figure 1.3, which divides the overall controller into three levels. The highest level includes tasks such as navigation, path planning and following, as well as obstacle avoidance. It condensates all these considerations into the desired movement of the rover, which is usually described as forward, lateral and yaw velocity. Computing the setpoint for each actuator from the desired rover velocity trajectories is subsequently done by the chassis control and since there are more actuation variables than control variables, infinitely many solutions exist. The resulting Degrees of Freedom (DoFs) can be used to optimize the locomotion behavior with respect to minimum energy consumption, maximum traction or other objectives, depending on current terrain and availability of sensor data. Problems with the crossing of sandfields, as described in the last paragraph, might be solvable and energy could be saved on easy terrain. The bottom layer in the control hierarchy of Figure 1.3 are the motor controllers, which, on an individual motor level, make sure that the setpoint rates, angles or torques are realized. These are also called servo drives and are usually implemented on separate hardware and executed with a much higher control frequency than the chassis and high level controllers.

1.3 State of the Art

The development of suitable synthesis models is one of the main challenges in designing a model-based chassis controller. For an exploration rover on rough terrain, the system behavior can be divided into the 3D multi-body dynamics, the redundant kinematics, the actuator dynamics and the wheel-ground interaction. Two subsystems have a particularly high impact on the system behavior. First, the over-actuation of the rover, since it allows for an optimization of the actuator commands with respect to various objectives. Second, the wheel-ground contact due to its significant influence on the locomotion, see Section 1.2.

Wheel-ground interaction

The interaction of objects like wheels with soft soil is called terramechanics and modeling it is a challenging field of research on its own. Depending on the application, models of different fidelity have been developed at our institute [Lic+16] and in other research groups. When it comes to fast models, that are suitable for usage in a multi-body simulation of a full rover system, semi-empirical models are commonly used. Most of the approaches are based on the work of Bekker, Wong, Reece, Janosi and Hanamoto and the foundation can be found in the book of Wong [Won08]. Many adaptations have been made, Ding et al. [Din+15b] provide a good overview of the field and published flaws of different approaches were summarized in [Bar18]. Independent of the actual model technique that is used, the coupling of terramechanics models and multi-body simulation – just as other penetration-based contact models – is always on the contact force level. Even more complex models like our in-house Soil Contact Model (SCM) [Bus18], which includes a 2-dimensional soil discretization and considers soil flow, ultimately provide a contact force that acts on the contact object. Due to this fact the chassis controller has to act on the force level at some point to consider the crucial effects of terramechanics.

An alternative to the consideration of wheel-ground contact effects in the chassis controller is to take them into account in the path planning already. The authors of [Hel+04; INY07], for example, incorporated a slip estimation in the path following algorithm, which corrects the rover velocity commands.

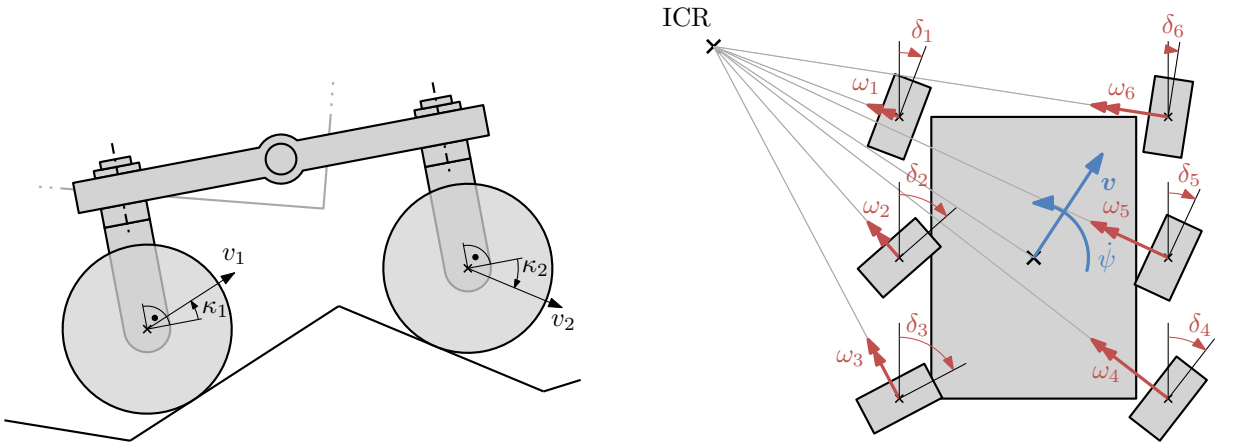
Over-actuated vehicle

The second emphasized feature of a rover is its over-actuation due to more actuated than control variables. Thus a thorough literature review of over-actuated systems in general and wheeled mobile robots in particular is given in this section. Modeling such an over-actuated system leads to an underdetermined system of equations, which for the linear and static case reads

$$\underline{\boldsymbol{A}}\boldsymbol{x} = \boldsymbol{b} \tag{1.1}$$

with the $(m \times n)$-matrix $\underline{\boldsymbol{A}}$ and $m < n$. For the $\underline{\boldsymbol{A}}$-matrix having full row rank, $\operatorname{rank}(\underline{\boldsymbol{A}}) = m$, there are infinitely many solutions for $\boldsymbol{x}$ [JF13]. Two different strategies can be found in the literature for the control of such a vehicle, see the top blue box of Figure 1.4. Additional constraints to make the problem determined, yielding a unique solution, can be introduced or the available DoFs might be used for an optimization of specific objectives.

Constraint-based solutions. This solution for the underdetermined problem is relatively easy, has low computational costs and yields explicit equations for the actuation variables. It was used in JPL's past and current planetary exploration rovers Spirit, Opportunity and Curiosity [Arv+16; BM06] as well as in the development of ESA's Rosalind Franklin rover [Bor+17]. The additional constraints are derived from an *Instantaneous Center of Rotation*



(a) The contact angle κ is defined as the angle between the forward direction on a level ground, i.e. 90° to the steering axis, and the actual velocity direction of the wheel hub.

(b) Geometric constraint-based control with the ICR for a six wheel all-wheel-steering robot. Blue are the input velocity and yaw rate and red are the actuator setpoints.

Figure 1.5: Sketches of contact angles (left) and the ICR (right).

(ICR) for the rover movement, see Figure 1.5b. Steering angles and rotation velocities of the wheels are then computed such that the latter all move longitudinally with matching speeds assuming zero wheel slip. This method is unambiguous if all wheel steering angles are restricted to a 180° sector and the rover kinematics are used to derive the explicit equations. Due to the geometric nature of the additional constraints, this method is also referred to as *geometric control* or *Ackermann* control, indicating the similarity to the steering geometry for road vehicles.

Besides its usage in exploration missions, the constraint-based approach is also applied in most research work that focuses on other parts of the controller such as path optimization, path following or autonomous navigation. For rough terrain, the translational movement of the wheels cannot be assumed to be in a plane, e.g. if one wheel climbs over a rock, it follows a different trajectory than the wheels on flat ground. To ensure a consistent speed in the desired rover direction, the rates of the wheels need to be adjusted, such that the translational velocity component in the rover movement direction has the same length. This is called *wheel-fighting* prevention and was first presented by Peynot and Lacroix [PL03] for their rover with what they call independently driven non-directional wheels. For the computation of the wheel rate adjustments, the contact angles must be estimated, which was first presented by Iagnemma and Dubowsky [ID00]. Different definitions of the contact angle can be found in literature, but this work follows the one from Iagnemma, who defines them as the angle κ_i between the wheel forward direction and the actual translational velocity vector v_i, see Figure 1.5a. Another algorithm to prevent wheel-fighting was later developed for the Curiosity rover, uploaded to the rover in 2017 and is used in its operations since then [Tou+18]. For a simulated 4-wheel rover, standard geometric control, slippage consideration and contact angles were recently combined to one controller [GI16]. It is based

on a ground property estimation and a switching algorithm between the different control strategies.

Optimal control allocation. To make optimal use of the over-actuation, the full solution space of the underdetermined system of equations (1.1) must be exploited, see the *find optimum* branch in Figure 1.4. All solutions of (1.1) can be decomposed into

$$\boldsymbol{x} = \underline{\boldsymbol{A}}^{+}\boldsymbol{b} + \underline{\boldsymbol{A}}^{\perp}\boldsymbol{\rho} \tag{1.2}$$

with the *generalized inverse* or *pseudo-inverse* $\underline{\boldsymbol{A}}^{+}$ and a base of the $\underline{\boldsymbol{A}}$-matrix's null-space $\underline{\boldsymbol{A}}^{\perp}$ [BG74]. While the first term of (1.2) represents the minimum norm solution of (1.1), the second term is its homogeneous solution and projects an arbitrary vector $\boldsymbol{\rho}$ into the null-space of the $\underline{\boldsymbol{A}}$-matrix. The intuitive interpretation is that $\boldsymbol{\rho}$ represents the $(n-m)$ DoFs of the inversion of (1.1), since $\boldsymbol{\rho}$ does not influence the right-hand side $\boldsymbol{b}$. This mathematical problem of solving an underdetermined linear system of equations with generalized inverses dates back to the early 1900's. A comprising analysis was published by Penrose, see [BG74] for more details. The first applications were made in the domain of kinematically redundant robots in the 1970's. Therein, the mapping between joint velocities and end effector velocity through the Jacobi-matrix yields a system of type (1.1), see the review paper of Klein and Huang [KH83]. An application to mobile robots was published in 1987 with the force allocation of a legged robot [KC87] and applications in the aeronautics sector are presented in [JF13; AKB06].

The underdetermined allocation (1.1) for over-actuated wheeled mobile robots can be stated on velocity-level, see the *velocity*-branch in the red box of Figure 1.4. This system of equations results from the differential kinematics of the robot and includes dependencies on chassis articulations, e.g. bogies. Tarokh [Tar16] implemented this for a rover simulation that includes a wheel rate controller to drive the rover. The authors of [WBK07] designed a controller that uses this velocity-based control allocation for a wheeled mobile robot with a manipulator on even ground. To consider terramechanics forces within a velocity-based optimization, Krenn et al. [Kre+13a] used a simplified soft soil model for the prediction of a model predictive controller.

Force-based optimal allocation for wheeled mobile robots. Most optimization-based approaches, however, use the force-torque equilibrium between contact forces and rover level forces to derive the underdetermined system of equations. Due to the need to consider the effects of the force-based wheel-ground contact, see the first paragraph of this section, the present work focuses on force-based allocation as well. Taking into account the velocity interface to the high-level controller that was introduced in Figure 1.3, a dynamic system results. Instead of using the resulting dynamic underdetermined system of equations for the controller synthesis directly, it is convenient to divide the controller into sub-controllers that are depicted in the green box in Figure 1.4 [JF13]. By introducing the same number of *virtual inputs* as the trajectory from the high-level controller, the dynamics and the

underdeterminacy of the total model can be handled separately. The first sub-controller, *system level* in Figure 1.4, computes the virtual input trajectory from the velocity input of the high-level controller. In the *allocation*, these virtual inputs are then distributed to the individual effectors, i.e. wheel units of the rover. Due to the separation into sub-controllers, the allocation remains a static problem and can be expressed as the underdetermined system of equations (1.1) in the linear case. The dynamics of the *effectors* can optionally be controlled by individual underlying sub-controllers They act on the effectors independently of the others and ultimately calculate the manipulated variable trajectories to be sent to the motor controllers.

The system-level controller for mobile robots is mostly not model-based but either performed by a PID-controller [ID04a] or not mentioned at all. Michaud et al. [Mic+08] compute the forces to maintain static stability model-based and only use a rover velocity PID-controller for changes in the movement of the rover.

For the force allocation, most approaches perform an optimization on the full static underdetermined system directly, see *optimization of full allocation* in Figure 1.4. The additional DoFs can be used to optimize different criteria, such as energy consumption [ID04b; Sil+10] or a stability measure that aims at preventing tip-over [MRM19]. For more complex systems, advanced objectives, like a desired impedance force at a manipulator that is mounted to the rover, can also be included [BDO18]. In [Sil+10], the allocation problem is first manually reduced by considering constraints directly, e.g. the maximum motor torque and maintaining ground contact. Due to the high uncertainty of ground contact force models, especially for soft soil, Iagnemma and Dubowsky [ID04a] and Lamon and Siegwart [LS05] compute the ratios of traction force and normal force per wheel and minimize their sum or sum of squares. Since the ground contact is based on friction, the traction force can be assumed proportional to the normal force. Minimizing said fraction therefore is in fact a minimization of the usage of the maximum traction force potential. In other words, the sum of squares of the wheels' margin to soil failure is maximized.

The second type of optimization, *least squares* in Figure 1.4, is employed by Lhomme-Desages, Grand, and Guinot [LGG07] and leads to a less computationally expensive controller. Looking at the underdetermined system (1.1) and its general solution (1.2), only the first term is used and either computed with the pseudo-inverse or a least squares algorithm. While this is a minimization of the norm of allocated forces, other objectives like the ones discussed above cannot be pursued.

As published e.g. in [HDB98; LSS03], the decomposition (1.2) of the general allocation solution can also be applied to the force balance for wheeled mobile robots. These authors use the solution of the allocation for modeling, analyzing and simulating their robot systems but not for a controller synthesis. Even beyond that, no publications of rover controllers that are based on this decomposition were found within the literature review for this work.

A hybrid approach was pursued in an earlier work of the author [BZ17], where the driving torques were allocated with an optimization algorithm, while the desired steering angles

were computed through an ICR, i.e. constraint-based, approach. Although this leads to coordinated steering angles and a rather simple optimization problem, it does not exploit the full solution space and thus cannot guarantee optimality.

Over-actuated road vehicles. In the article of Krenn et al. [Kre+13b], control approaches for planetary exploration rovers and robotic road vehicles were shown in comparison. Both systems have individual wheel drive and steering and are controlled with the three velocity inputs named in Figure 1.3. An over-actuation is thus a common feature of both systems and the control and actuation variables show large overlap. When it comes to the synthesis model, differences arise however: While rovers move slowly on rough and partly sandy terrain, which necessitates complex ground contact models and non-planar movement, more detailed dynamics and tire models are needed for road vehicles. The control objectives for a rover are mostly energy consumption and traction capabilities, whereas passenger safety and ride comfort play a major role for road vehicles.

Nevertheless, it is worth investigating approaches from the field of over-actuated road vehicles, since they use the decomposition into minimum norm and homogeneous solution from (1.2) and the bottom right in the overview Figure 1.4. A controller for a general vehicle with individually driven and steered wheels, based on the vehicle's nonlinear dynamics, is developed in [Ore04], extended with wheel load distribution in [Ore05] and its performance is compared to a standard vehicle in multiple scenarios in [Ore06]. Knobel, Pruckner, and Bünte [KPB06] consider actuator rate limits for the optimization and Bünte and Andreasson [BA06] linearize a hybrid optimization objective to be able to solve it in real-time with a least-squares algorithm. Different drive topologies for such an over-actuated vehicle were studied by Hoedt [Hoe13], including investigations of fault tolerance and robustness as well as the design of the disturbance behavior. An investigation of such vehicle controls in the presence of disturbances and model uncertainties was published in [Mos16] and efforts were made to cope with challenges of the implementation on an actual vehicle.

All mentioned approaches from the field of over-actuated road vehicles use a model-based controller for the system level sub-controller to generate the virtual input trajectory.

1.4 Objectives and Approach

The overall objective of this work is to design a chassis controller for the ExoMars rover as an example of a six-wheeled mobile robot with individual wheel drive and steering. Rover velocity trajectories ought to be the input to this control system and commands for each actuator have to be computed. Before detailing the approach of this work, the overall goals for the chassis controller design are defined:

Obj 1 A fully model-based controller is to be designed to be able to assess the control performance benefits of using model knowledge.

Obj 2 The crucial effects of the wheel-ground contact behavior, which is commonly modeled as contact forces, must be considered and the approach shall allow for an easy adaptation of the model.

Obj 3 The approach has to exploit the full solution space of the over-actuated rover with respect to variable objectives.

Obj 4 For future use on onboard computers, the availability of sensors and the computational effort of the controller should be kept in mind.

To be able to optimize the locomotion behavior (Obj 3) and consider contact forces (Obj 2), a force-based allocation is used instead of a constraint-based approach. This results in the general controller structure that was introduced in the green box in Figure 1.4. As opposed to the existing contributions that were listed for wheeled mobile robots in Section 1.3, model-based controllers (Obj 1) are designed for each of the three sub-controllers *system level*, *allocation* and *effector*.

The non-planar, three dimensional movement in the gravity field as well as simplified dynamics are considered on system level. The articulated kinematics of the rover are used to derive the force allocation equation and a minimization of the traction potential usage is pursued. In such a cascaded approach, all underlying controllers need to be able to achieve any trajectory that is computed by a higher level sub-controller. For the present controller structure, this results in the need to design the system level and allocation models with virtual inputs. These must be defined such that, first, the dynamic system level controller yields a unique solution and, second, the allocation equation has full row rank and is thus solvable. Finding such suitable virtual inputs and modifying the sub-models accordingly is one key aspect of the controller design in this work.

Having modified the models, the decomposition of the allocation equation's general solution, as in (1.2), allows for a beneficial combination of computational efficiency (Obj 4) while using the full solution space (Obj 3). This can be achieved by performing the optimization of the DoFs $\boldsymbol{\rho}$ at a much slower update rate than the rest of the controller, which always uses the most recent available set of DoFs. Due to $\boldsymbol{\rho}$ being projected into the null-space of the allocation matrix, it has no impact on the system level forces and torques. The force-torque equilibrium is thus always fulfilled, irrespective of the choice of DoFs, only the optimality of the solution with respect to the objective function can be compromised. The overall computation cost is thereby reduced without affecting the optimality too much, since the influencing rover states do not change rapidly. Such a control approach was not found within the field of wheeled mobile robot control, the controller structure is thus more similar to the domain of over-actuated road vehicles, see Section 1.3. The specifics of rough terrain, kinematics, passive chassis, slow movement and ground contact properties, on the other hand, are more related to the listed publications from the mobile robot domain.

An effector sub-controller finally takes actuator dynamics of steering and drive motors into account and combines this with contact force laws. Thereby, the contact properties at indi-

vidual wheel level are handled by the underlying effector controller. At the same time, the optimization ensures the system-wide optimal usage of the effector traction potentials by adjusting the allocation DoFs (Obj 2).

This comprising model development and the modifications to the models for a rigorous chassis controller synthesis have not been done for wheeled mobile robots to the authors knowledge and are the main contribution of this work. The overall controller structure and a brief summary of models and synthesis was recently published in the journal *at - Automatisierungstechnik* [BK20].

1.5 Assumptions

The assumptions that are made for the model and controller design are briefly summarized in this section and rationales are given.

- *One contact point per wheel* is assumed, which only transfers forces in all directions. The torques on rover level hence purely results from the contact forces with their lever arms.
- *Resistance force* is not modeled explicitly since it is highly dependent on the ground type and sinkage state, which are both not measurable. Research is currently done at our institute on real-time capable modeling of resistance forces and more accurate traction force models for future development of the chassis control, see Section 3.6. However, in the present work the resistance force is an unknown disturbance that needs to be compensated for by the integral action of the controller.
- *Moderately rough terrain* leads to the assumption of zero contact angles, which means that the wheels always move in the plane that is orthogonal to the steering axis, see Figure 1.5a. Moderate terrain changes are reflected in considering the bogie angles, whereas driving over a sharp edge or rock is not accounted for by the model and must thus be handled by the integrator part of the controller. A comment on existing work on contact angles, first efforts at our institute and possible future work in that direction is given in the outlook in Section 6.2.
- *Ideally measurable* linear and angular rover velocity, rover orientation, bogie angles, steering rates and angles, wheel rates as well as motor torques are assumed. An estimation of the rover velocities and orientation currently relies on visual odometry, potentially fused with rate and acceleration sensors. On earth, the ability of algorithms fusing visual odometry, acceleration and angular velocity from an Inertial Measurement Unit (IMU) and wheel odometry to provide the desired quantities in real-time has been shown e.g. by Schmid [Sch15]. While these estimates are not computable in real-time on past and present planetary rovers, a reasonable update frequency of 0.1 Hz is planned for the ExoMars Rosalind Franklin rover already [Bor+17] and will evolve further with more onboard computational power and refined algorithms.

1.6 Structure of the Thesis

In Chapter 2, the notation systematics is explained and concepts of model development and controller synthesis, that are used in this work, are briefly depicted. This discourse is not comprehensive, since its only purpose is to allow for a controls or robotics engineer to read the thesis without needing to study the given references.

The development of the synthesis models is shown in Chapter 3, starting with a presentation of the ExoMars Breadboard 2 (BB2) rover that is being considered in this work. The different sub-models are derived in separate sections, first, the rover kinematics in Section 3.2 and its dynamics in Section 3.3 as well as the wheel dynamics in Section 3.4. For a better understanding of terramechanics, Section 3.5 gives an overview of modeling techniques and details on the most common approach for full system simulations, the semi-empirical models. In Section 3.6, a novel empirical and dynamic terramechanics model for time-critical applications is developed, before introducing the simplified synthesis contact model in Section 3.7. The latter model and the wheel dynamics purely act on the single wheel layer without interaction with the other wheels, thus they are combined into one dynamic model in Section 3.8, before a summary of the whole rover model is provided in Section 3.9.

Chapter 4 shows the controller design for the ExoMars BB2 rover. Upfront, the virtual inputs of the controller, see Figure 1.4, are derived and model modifications to the system level and the allocation model are made in Section 4.1. The three sub-controllers are subsequently designed, starting with a feedback linearization controller for the system level in Section 4.2 and an analysis of the stability of its internal dynamics. Secondly, the resulting rover forces are allocated with a decomposition of the general solution into minimum norm and homogeneous part and the optimization of the DoFs is derived in Section 4.3. In Section 4.4, an underlying wheel-ground dynamics controller is finally synthesized with dynamic extension and feedback linearization to ensure the desired contact forces for the movement of the rover.

The presented chassis control approach is verified with a co-simulation of the controller and a high-fidelity rover model in Chapter 5. After briefly describing the verification model and the simulation setup in Section 5.1, its performance is shown in two different scenarios. The nominal performance is compared to an ICR-based controller in Section 5.2 with only minor modeling inaccuracies. Therefore, the synthesis ground contact model is used without modifications in the high-fidelity simulation. The robustness of the control system to disturbances is shown in Section 5.3, before analyzing and discussing performance and potentials of the approach in Section 5.4.

In Chapter 6, the setting, approach and results of this work are summarized and topics for further work are highlighted.

2 Fundamentals

It is attempted in this work to provide an understandable and complete dissertation of the topic while keeping the explanations as tidy as possible. In that context, this second chapter aims at providing the necessary fundamentals of the different areas that are touched by this thesis. After reading this chapter, the experienced robotics or controls engineer should be able to follow the modeling and controller synthesis chapters without needing to consult additional literature.

First and foremost, Section 2.1 clarifies the notation systematics that is used consistently throughout this work. The short introductions of kinematics in Section 2.2 as well as of differential kinematics and statics in Section 2.3 follow closely the robotics book of Siciliano et al. [Sic+10]. From the area of system dynamics and controls, the theory of feedback linearization and the Lyapunov stability of nonlinear systems are briefly captured in Sections 2.4 and 2.5, respectively. Section 2.6 gives an overview of state and output controllability of linear and nonlinear systems. Lastly, Section 2.7 introduces the field of control allocation, which might be the least common and therefore most interesting of the topics covered in this chapter.

2.1 Notation

A consistent notation, that is used in the remainder of this thesis, is introduced in this section. Vectors $\boldsymbol{v}/\boldsymbol{V}$ are written in bold letters and matrices $\underline{\boldsymbol{M}}/\underline{\boldsymbol{m}}$ in bold and underlined letters. Indices are written italic if they are variables themselves, e.g. the x indicating the x-direction in the velocity component v_x. All other indices, e.g. identifiers such as in a maximum force F_{max}, are written upright. Additional model parameters are mostly given the letter k with a describing index to easily distinguish them from dynamic variables.

The biggest need for a strict convention, however, comes from the expression of quantities in different coordinate systems (also called *frames*) and the rotation between them. Position vectors are noted with the bold letter ${}^{\mathrm{c}}\boldsymbol{r}_{\mathrm{b|a}}$, where the lower right index consists, from left to right, of the tip of the vector, a vertical bar and its base. The describing frame is indicated as the upper left index and

$$ {}^{\mathrm{c}}\boldsymbol{r}_{\mathrm{b|a}} \tag{2.1} $$

is thus the vector from 'a' to 'b', described in frame 'c'. If the describing frame is equivalent to the base of the vector, the upper left index can be omitted,

$$ {}^{\mathrm{a}}\boldsymbol{r}_{\mathrm{b|a}} = \boldsymbol{r}_{\mathrm{b|a}} \, . \tag{2.2} $$

A component of the vector is written with the component number or direction, e.g. x, in the lower right index after a comma,

$$ {}^{\mathrm{c}}r_{\mathrm{b|a},x} \ . \tag{2.3} $$

The component is always selected in the describing frame, i.e. the frame 'c' in this case.

2.2 Kinematics

As said before, the following subsections are close to the book of SICILIANO et al. [Sic+10] and the interested reader is referred to this book for further details. The section starts with some clarifications on rotation matrices in Section 2.2.1 and extends to rotations in 3D in Section 2.2.2, introducing the Euler angles. Rotation and translation are combined to the homogeneous transformation matrix in Section 2.2.3, which is then chained to form the direct kinematics with joint variables in Section 2.2.4. DENAVIT-HARTENBERG's concept to systematically derive the homogeneous transformations for any robotic system is finally shown in Section 2.2.5, since it is applied to the rover in this work.

2.2.1 Rotation Matrix

Each rotation matrix ${}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}}$, in this case from 'a' to 'b', must have a determinant of 1,

$$ \det(\underline{\boldsymbol{R}}) = 1 \ , \tag{2.4} $$

and must be orthogonal $\underline{\boldsymbol{R}}^{\top}\underline{\boldsymbol{R}} = \underline{\boldsymbol{I}}$, which leads to the useful property

$$ {}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}}^{-1} = {}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}}^{\top} = {}^{\mathrm{b}}\underline{\boldsymbol{R}}_{\mathrm{a}} \ . \tag{2.5} $$

Three interpretations of a rotation matrix are summarized in the following, since their use and direction of rotation is prone to making mistakes:

- *Unit vectors of a coordinate system* 'b' *in system* 'a': The columns of the rotation matrix ${}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}} = [{}^{\mathrm{a}}\boldsymbol{e}_{\mathrm{b}x}, {}^{\mathrm{a}}\boldsymbol{e}_{\mathrm{b}y}, {}^{\mathrm{a}}\boldsymbol{e}_{\mathrm{b}z}]$ represent the unit vectors $\boldsymbol{e}$ of a coordinate system b, resolved in frame a. This also marks a straightforward way to derive a rotation matrix between coordinate systems if the unit vectors of one of them, described in the other frame, are known.

- *Change description system for vector from* 'b' *to* 'a': The change of the describing coordinate system of a vector $\boldsymbol{r}$ can be achieved with the rotation matrix ${}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}}$ as well,

  $$ {}^{\mathrm{a}}\boldsymbol{r} = {}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}}\,{}^{\mathrm{b}}\boldsymbol{r} \ . \tag{2.6} $$

  To change the describing frame from 'b' to 'a', property (2.5) can be used to get to

  $$ {}^{\mathrm{b}}\boldsymbol{r} = {}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}}^{\top}\,{}^{\mathrm{a}}\boldsymbol{r} = {}^{\mathrm{b}}\underline{\boldsymbol{R}}_{\mathrm{a}}\,{}^{\mathrm{a}}\boldsymbol{r} \ . \tag{2.7} $$

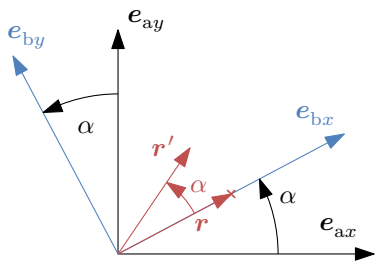


Figure 2.1: 2D example that shows the different interpretations of a rotation matrix.

- *Rotate a vector*: If the describing coordinate system is kept identical, multiplying a rotation matrix to a vector rotates the vector, similar to the unit vectors in the first item of this list.

Within this work, rotation matrices are mostly used for changing the describing frame of a vector.

A simple 2-dimensional example to clarify the different meanings of a rotation matrix can be seen in Figure 2.1 with

$$ {}^{a}\underline{\boldsymbol{R}}_{b} = \begin{bmatrix} {}^{a}\boldsymbol{e}_{bx} & {}^{a}\boldsymbol{e}_{by} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}. \tag{2.8} $$

By using the first interpretation of the above list, the rotation matrix ${}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}}$ was built from the unit vectors of frame 'b', described in frame 'a'. With the second interpretation from above, vector $\boldsymbol{r}$ in Figure 2.1 can be transformed from its representation in system 'b' to system 'a',

$$ {}^{\mathrm{a}}\boldsymbol{r} = {}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}}\, {}^{\mathrm{b}}\boldsymbol{r} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} {}^{\mathrm{b}}\begin{bmatrix} 0.5 \\ 0 \end{bmatrix} = 0.5 \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix}. \tag{2.9} $$

Finally, by keeping the describing coordinate system 'b', the matrix-vector multiplication

$$ {}^{\mathrm{b}}\boldsymbol{r}' = {}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}}\, {}^{\mathrm{b}}\boldsymbol{r} \tag{2.10} $$

results in a rotation of the vector $\boldsymbol{r}$ about the angle α, see $\boldsymbol{r}'$ in Figure 2.1.

2.2.2 Rotation in the 3-Dimensional Space

For the 3-dimensional space, rotation matrices $\underline{\boldsymbol{R}} \in \mathbb{R}^{3\times3}$ about the three axes of a cartesian coordinate system can be defined as

$$\underline{\boldsymbol{R}}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}, \tag{2.11}$$

$$\underline{\boldsymbol{R}}_y(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}, \tag{2.12}$$

$$\underline{\boldsymbol{R}}_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.13}$$

An arbitrary rotation can be described by subsequent rotations about suitable axes, meaning that two subsequent rotation axes must not be parallel to one another. This results in 12 possible rotation sequences that are represented by their triplet of angles. Depending on the author and field, all 12 possible combinations, e.g. [Sic+10], or only a subset of them, e.g. [SE17], is denoted as *Euler angles*. One common split of the 12 valid sequences is between the six possible sequences with *proper Euler angles* and the six ones with the *Tait-Bryan angles* or *Cardan angles*. The former are all sequences with the first and third rotation about the same axis, e.g. x-y-x, where the second and third rotation can either be about the respective intermediate axis of the last elemental rotation or about the original one. Tait-Bryan or Cardan angles, on the other side, are the group of sequences that use all three elemental rotations. For vehicles, the *Roll-Pitch-Yaw (RPY)* convention from the group of Cardan angles is commonly used. The fixed '0'-frame (x, y, z) is first rotated about its x-axis by the roll angle ϕ, resulting in the one-dash intermediate system (x', y', z'). Secondly, the one-dash system is rotated about its y'-axis by the pitch angle θ, yielding the two-dash intermediate system (x'', y'', z''). The final '1'-frame is now obtained by rotating the two-dash system about its z''-axis by the yaw angle ψ. This results in the total rotation matrix

$$\underline{\boldsymbol{R}}_{\mathrm{RPY}}(\phi, \theta, \psi) = {}^0\underline{\boldsymbol{R}}_1 = \underline{\boldsymbol{R}}_z(\psi)\underline{\boldsymbol{R}}_y(\theta)\underline{\boldsymbol{R}}_x(\phi) \tag{2.14}$$

from a fixed reference frame '0' to the current frame '1'. According to (2.9), the resulting rotation matrix can be used to change the describing coordinate system of a vector $\boldsymbol{r}$ as

$${}^0\boldsymbol{r} = \underline{\boldsymbol{R}}_{\mathrm{RPY}}\,{}^1\boldsymbol{r}\,, \tag{2.15}$$

$${}^1\boldsymbol{r} = \underline{\boldsymbol{R}}_{\mathrm{RPY}}^{\top}\,{}^0\boldsymbol{r}\,. \tag{2.16}$$

2.2.3 Homogeneous Transformations

Sections 2.2.1 and 2.2.2 consider a pure rotation, where the origins of the systems $\boldsymbol{r}_{\mathrm{a}} = \boldsymbol{r}_0$ are identical. If the origins are not congruent, see Figure 2.2, a vector $\boldsymbol{r}_{\mathrm{p|a}}$ pointing from the

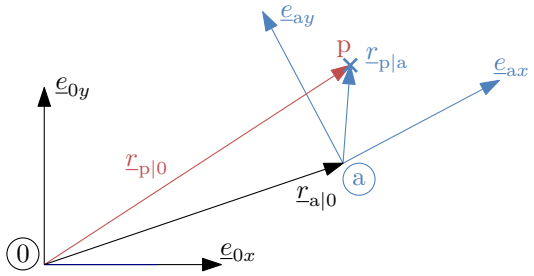


Figure 2.2: Describing position vectors from different coordinate systems.

origin of the system 'a' to a point 'p' can be transformed to frame '0' with

$$\boldsymbol{r}_{\mathrm{p|0}} = {}^{0}\underline{\boldsymbol{R}}_{\mathrm{a}}\boldsymbol{r}_{\mathrm{p|a}} + \boldsymbol{r}_{\mathrm{a|0}} \,. \tag{2.17}$$

For a tidy notation, the *homogeneous transformation matrix* ${}^{0}\underline{\boldsymbol{T}}_{\mathrm{a}}$ is defined such that

$$\begin{bmatrix} \boldsymbol{r}_{\mathrm{p|0}} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} {}^{0}\underline{\boldsymbol{R}}_{\mathrm{a}} & \boldsymbol{r}_{\mathrm{a|0}} \\ \boldsymbol{0}^{\top} & 1 \end{bmatrix}}_{={}^{0}\underline{\boldsymbol{T}}_{\mathrm{a}}} \begin{bmatrix} \boldsymbol{r}_{\mathrm{p|a}} \\ 1 \end{bmatrix} . \tag{2.18}$$

Homogeneous transformation matrices can be multiplied for a sequence of transformations analogously to the rotation matrices

$${}^{0}\underline{\boldsymbol{T}}_{\mathrm{b}} = {}^{0}\underline{\boldsymbol{T}}_{\mathrm{a}}\,{}^{\mathrm{a}}\underline{\boldsymbol{T}}_{\mathrm{b}} = \begin{bmatrix} {}^{0}\underline{\boldsymbol{R}}_{\mathrm{a}}\,{}^{\mathrm{a}}\underline{\boldsymbol{R}}_{\mathrm{b}} & {}^{0}\underline{\boldsymbol{R}}_{\mathrm{a}}\boldsymbol{r}_{\mathrm{b|a}} + \boldsymbol{r}_{\mathrm{a|0}} \\ \boldsymbol{0}^{\top} & 1 \end{bmatrix} . \tag{2.19}$$

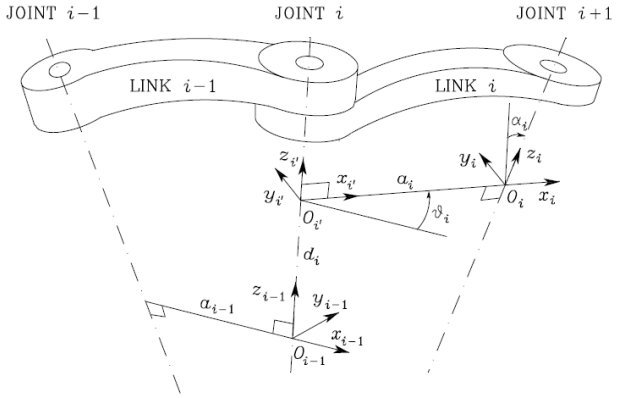
However, the inverse transformation matrix is not equal to its transpose

$${}^{\mathrm{a}}\underline{\boldsymbol{T}}_{0} = {}^{0}\underline{\boldsymbol{T}}_{\mathrm{a}}^{-1} = \begin{bmatrix} {}^{0}\underline{\boldsymbol{R}}_{\mathrm{a}}^{\top} & -{}^{0}\underline{\boldsymbol{R}}_{\mathrm{a}}^{\top}\boldsymbol{r}_{\mathrm{a|0}} \\ \boldsymbol{0}^{\top} & 1 \end{bmatrix} \neq {}^{0}\underline{\boldsymbol{T}}_{\mathrm{a}}^{\top} \,. \tag{2.20}$$

2.2.4 Direct Kinematics and Joint Variables

A series of rigid bodies (*links*) and prismatic and revolute *joints* forms a *kinematic chain* with n DoFs. The DoFs are called *joint variables* and commonly given the symbol q which can be both, an angle or a distance, depending to the joint type. Note that the term DoF denotes actual free variables of the over-actuated system in this work, therefore, it is avoided in the remainder to use DoF in the context of joint variables. The homogeneous transformation from the base 'ba' to the end 'e' of a kinematic chain ${}^{\mathrm{ba}}\underline{\boldsymbol{T}}_{\mathrm{e}}$ can be expressed as a series of rotations and translations ${}^{i-1}\underline{\boldsymbol{T}}_{i}(q_i)$ for $i = 1, \ldots, n$. Linking the transformations with (2.19), the total transformation matrix is dependent on the vector of joint variables $\boldsymbol{q} \in \mathbb{R}^{(n\times 1)}$ and has the form

$${}^{\mathrm{ba}}\underline{\boldsymbol{T}}_{\mathrm{e}}(\boldsymbol{q}) = \begin{bmatrix} {}^{\mathrm{ba}}\underline{\boldsymbol{R}}_{\mathrm{e}}(\boldsymbol{q}) & \boldsymbol{r}_{\mathrm{e|ba}}(\boldsymbol{q}) \\ \boldsymbol{0}^{\top} & 1 \end{bmatrix} . \tag{2.21}$$



(a) DH kinematic parameters (from [Sic+10]).

Link	a_i	α_i	d_i	ϑ_i
1	a_1	α_1	d_1	ϑ_1
...	...	...	...	...
n	a_n	α_n	d_n	ϑ_n

(b) DH parameter table for a kinematic chain with n joint variables.

Figure 2.3: The DH parameter concept and the schematics of a DH parameter table.

2.2.5 Denavit-Hartenberg Convention

The Denavit-Hartenberg (DH) convention is a systematics for creating the direct kinematics of an open kinematic chain. For each link i of the chain, a coordinate system is defined following certain rules that can be found in [Sic+10, p. 61]. Subsequently, n transformation matrices between link $(i-1)$ and link i are computed, where each depends on four parameters, which are determined by the choice of the frames. The procedure and the meaning of the parameters can be understood best by looking at Figure 2.3a and following four steps to perform the transformation from frame $(i-1)$ over the intermediate frame i' to i:

1. Rotate the system by ϑ_i about the z_{i-1} axis to get to the vectors $x_{i'}$, $y_{i'}$ and $z_{i'}$.
2. Translate the system by d_i along the z_{i-1} axis to get to the origin $O_{i'}$.
3. Translate the system by a_i along the $x_{i'}$ axis to get to the origin O_i.
4. Rotate the system by α_i about the $x_{i'}$ axis to get to the vectors x_i, y_i and z_i.

The rotation (ϑ_i)/translation (d_i) axis of the ith joint is always the z_{i-1} axis, whereas the origins of the frames are not on the joints or links in general. The homogeneous transformation matrix between two frames results as

$$ {}^{i-1}\underline{\boldsymbol{T}}_i(q_i) = \begin{bmatrix} \cos(\vartheta_i) & -\sin(\vartheta_i)\cos(\alpha_i) & \sin(\vartheta_i)\sin(\alpha_i) & a_i\cos(\vartheta_i) \\ \sin(\vartheta_i) & \cos(\vartheta_i)\cos(\alpha_i) & -\cos(\vartheta_i)\sin(\alpha_i) & a_i\sin(\vartheta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.22} $$

where the joint variable q_i is the angle ϑ_i for a revolute or the distance d_i for a prismatic joint. The complete kinematic chain is commonly represented by a DH parameter table as in Figure 2.3b and the individual transformation matrices can be multiplied as in (2.19) to obtain the full transformation from the base to the end of the kinematic chain ${}^{\mathrm{ba}}\underline{\boldsymbol{T}}_{\mathrm{e}}(\boldsymbol{q})$ from (2.21).

2.3 Differential Kinematics and Statics

Similarly to the last section, the overview of differential kinematics and statics follows closely the book of SICILIANO et al. [Sic+10] as well. If frame '0' in Figure 2.2 is fixed, whereas frame 'a' and point 'p' are time-dependent, i.e. move in space, the absolute position of point 'p' can be expressed as

$$\boldsymbol{r}_{\mathrm{p|0}}(t) = \boldsymbol{r}_{\mathrm{a|0}}(t) + {}^{0}\underline{\boldsymbol{R}}_{\mathrm{a}}(t)\boldsymbol{r}_{\mathrm{p|a}}(t) \; . \tag{2.23}$$

Remember that the upper left index that notes the describing system is omitted if it is the same as the base of the vector, see (2.2). To compute the total velocity of point 'p', (2.23) is differentiated with respect to time, yielding

$$\frac{\mathrm{d}\boldsymbol{r}_{\mathrm{p|0}}(t)}{\mathrm{d}t} = \dot{\boldsymbol{r}}_{\mathrm{a|0}} + {}^{0}\underline{\boldsymbol{R}}_{\mathrm{a}}\dot{\boldsymbol{r}}_{\mathrm{p|a}} + {}^{0}\dot{\underline{\boldsymbol{R}}}_{\mathrm{a}}\boldsymbol{r}_{\mathrm{p|a}} \; . \tag{2.24}$$

The first term is the velocity of frame 'a' and the second is the relative velocity of point $\boldsymbol{r}_{\mathrm{p}}$ with respect to frame 'a', rotated into the inertial frame to allow for the summation. The last term contains the time derivative of the rotation matrix from '0'- to 'a'-frame, which is better known as the cross product of the total angular velocity of frame 'a' $\boldsymbol{\omega}_{\mathrm{a|0}}$ with the distance $\boldsymbol{r}_{\mathrm{p|a}}$,

$${}^{0}\dot{\underline{\boldsymbol{R}}}_{\mathrm{a}}\boldsymbol{r}_{\mathrm{p|a}} = \boldsymbol{\omega}_{\mathrm{a|0}} \times {}^{0}\boldsymbol{r}_{\mathrm{p|a}} = \boldsymbol{\omega}_{\mathrm{a|0}} \times \left({}^{0}\underline{\boldsymbol{R}}_{\mathrm{a}}\boldsymbol{r}_{\mathrm{p|a}}\right) = \underline{\boldsymbol{S}}(\boldsymbol{\omega}_{\mathrm{a|0}}){}^{0}\underline{\boldsymbol{R}}_{\mathrm{a}}\boldsymbol{r}_{\mathrm{p|a}} \; . \tag{2.25}$$

Therein the skew-symmetric cross product building matrix is defined as

$$\underline{\boldsymbol{S}}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} . \tag{2.26}$$

A property, which will be required later, is that a rotation $\underline{\boldsymbol{R}}$ of the argument of the skew-symmetric $\underline{\boldsymbol{S}}$-matrix can be resolved following

$$\underline{\boldsymbol{S}}(\underline{\boldsymbol{R}}\boldsymbol{\omega}) = \underline{\boldsymbol{R}}\underline{\boldsymbol{S}}(\boldsymbol{\omega})\underline{\boldsymbol{R}}^{\top}. \tag{2.27}$$

Coming back to the open kinematic chain with the joint coordinates $\boldsymbol{q}$ from Section 2.2.4, the linear velocity of the end of the chain, with respect to the base, is expressed as

$$\dot{\boldsymbol{r}}_{\mathrm{e|ba}}(\boldsymbol{q}(t)) = \underbrace{\frac{\partial \boldsymbol{r}_{\mathrm{e|ba}}(\boldsymbol{q})}{\partial \boldsymbol{q}}}_{\underline{\boldsymbol{J}}_{v}} \dot{\boldsymbol{q}} \; . \tag{2.28}$$

The position vector $\boldsymbol{r}_{\mathrm{e|ba}}(\boldsymbol{q})$ is part of the homogeneous transformation matrix in (2.21) and its partial derivative yields the linear part $\underline{\boldsymbol{J}}_{v}$ of the *geometric Jacobian*. The angular velocity of the end frame results in a simple summation of the joint angular velocities,

$${}^{\mathrm{ba}}\boldsymbol{\omega}_{\mathrm{e}} = \sum_{i=1}^{n} {}^{\mathrm{ba}}\boldsymbol{\omega}_{i-1,i} \; . \tag{2.29}$$

Equation (2.29) can be expressed in terms of the joint rates $\dot{\boldsymbol{q}}$ and combined with (2.28) to define the full geometric Jacobian $\underline{\boldsymbol{J}}(\boldsymbol{q})$ with

$$\begin{bmatrix} \dot{\boldsymbol{r}}_{\mathrm{e}} \\ \boldsymbol{\omega}_{\mathrm{e}} \end{bmatrix} = \underline{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \ . \tag{2.30}$$

Through the principle of virtual work, the so-called *kineto-statics duality* [Sic+10] can be established with the transposed Jacobian,

$$\boldsymbol{\tau} = \underline{\boldsymbol{J}}^{\top}(\boldsymbol{q}) \begin{bmatrix} \boldsymbol{F}_{\mathrm{e}} \\ \boldsymbol{T}_{\mathrm{e}} \end{bmatrix} . \tag{2.31}$$

This relates the end effector forces $\boldsymbol{F}_{\mathrm{e}}$ and torques $\boldsymbol{T}_{\mathrm{e}}$ to the generalized joint forces and torques $\boldsymbol{\tau}$. Note that in the 3-dimensional space there are three forces and three torques and thus (2.31) is determined for six 1-DoF joints and underdetermined for more than that, assuming no rank deficiency of $\underline{\boldsymbol{J}}^{\top}$.

2.4 Feedback Linearization

This summary of feedback linearization control for Multi-Input Multi-Output (MIMO) systems is based on the nonlinear control systems book of Isidori [Isi95]. Feedback linearization of a MIMO system is a straightforward extension of the SISO case if the following can be established for the MIMO system

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \underline{\boldsymbol{g}}(\boldsymbol{x})\boldsymbol{u} \ , \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}) \ , \end{aligned} \tag{2.32}$$

with n_x states, n_u inputs and n_y outputs where

$$\begin{aligned} \boldsymbol{u} &= \begin{bmatrix} u_1 & \dots & u_{n_u} \end{bmatrix}^{\top} , \\ \boldsymbol{y} &= \begin{bmatrix} y_1 & \dots & y_{n_y} \end{bmatrix}^{\top} , \\ \boldsymbol{f}(\boldsymbol{x}) &= \begin{bmatrix} f_1(\boldsymbol{x}) & \dots & f_{n_x}(\boldsymbol{x}) \end{bmatrix}^{\top} , \\ \boldsymbol{h}(\boldsymbol{x}) &= \begin{bmatrix} h_1(\boldsymbol{x}) & \dots & h_{n_y}(\boldsymbol{x}) \end{bmatrix}^{\top} \end{aligned} \tag{2.33}$$

are column vectors and

$$\underline{\boldsymbol{g}}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{g}_1(\boldsymbol{x}), & \dots, & \boldsymbol{g}_{n_u}(\boldsymbol{x}) \end{bmatrix} \tag{2.34}$$

is an $n_x \times n_u$-matrix.

Lemma 1 *The system* (2.32) *is of vector relative degree* $\boldsymbol{p} = [p_1, \dots, p_{n_y}]^{\top}$ *if the matrix*

$$\underline{\boldsymbol{V}} = \begin{bmatrix} L_{\boldsymbol{g}_1} L_{\boldsymbol{f}}^{p_1-1} h_1(\boldsymbol{x}) & \dots & L_{\boldsymbol{g}_{n_u}} L_{\boldsymbol{f}}^{p_1-1} h_1(\boldsymbol{x}) \\ \vdots & \ddots & \vdots \\ L_{\boldsymbol{g}_1} L_{\boldsymbol{f}}^{p_{n_y}-1} h_{n_y}(\boldsymbol{x}) & \dots & L_{\boldsymbol{g}_{n_u}} L_{\boldsymbol{f}}^{p_{n_y}-1} h_{n_y}(\boldsymbol{x}) \end{bmatrix} \tag{2.35}$$

has full rank and

$$L_{\boldsymbol{g}_{i_u}} L_{\boldsymbol{f}}^{k} h_{i_y}(\boldsymbol{x}) \begin{cases} = 0 & \forall k < p_{i_y} - 1, 1 \leq i_u \leq n_u, 1 \leq i_y \leq n_y \\ \neq 0 & \forall k = p_{i_y} - 1, 1 \leq i_y \leq n_y \\ & \quad \textit{for at least one} \quad 1 \leq i_u \leq n_u \end{cases} . \tag{2.36}$$

Therein the *Lie-derivative*

$$L_{\boldsymbol{f}} h_{i_y}(\boldsymbol{x}) = \frac{\partial h_{i_y}(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) \tag{2.37}$$

and the kth Lie-derivative

$$L_{\boldsymbol{f}}^{k} h_{i_y}(\boldsymbol{x}) = \frac{\partial \left(L_{\boldsymbol{f}}^{k-1} h_{i_y}(\boldsymbol{x}) \right)}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) \tag{2.38}$$

were used. Descriptively, this means that each of the outputs is derived with respect to time until any of the inputs remains with a non-zero factor in the p_{i_y}th derivative. The number p_{i_y} is the relative degree of this output, provided the $\underline{\boldsymbol{V}}$-matrix has full rank. To compute $\underline{\boldsymbol{V}}$, each row is composed of the factors of the n_u inputs in the p_{i_y}th derivative of the i_yth output. As an example, if the first and second derivative of the first output of a system do not yield any input dependency and the third derivative is of the form

$$\overset{(3)}{y}_1 = \nu(\boldsymbol{x}) + 0u_1 + x_1 u_2 \, , \tag{2.39}$$

then the first row of $\underline{\boldsymbol{V}}$ is $[0, x_1]$. The system is assumed to have two inputs and $\nu(\boldsymbol{x})$ is an arbitrary function of the states. The same procedure is followed for all outputs and if the resulting $\underline{\boldsymbol{V}}$-matrix has full rank, the first output has a relative degree of three.

If Lemma 1 holds true, $p = \sum_{i=1}^{n_y} p_i$ new coordinates

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}_1 \\ \vdots \\ \boldsymbol{\xi}_{n_y} \end{bmatrix} = \begin{bmatrix} \gamma_1(\boldsymbol{x}) \\ \vdots \\ \gamma_p(\boldsymbol{x}) \end{bmatrix} \quad \text{with} \quad \boldsymbol{\xi}_i = \begin{bmatrix} L_{\boldsymbol{f}}^{0} h_i(\boldsymbol{x}) \\ \vdots \\ L_{\boldsymbol{f}}^{p_i - 1} h_i(\boldsymbol{x}) \end{bmatrix} \quad \text{for} \quad i = 1 \ldots n_y \tag{2.40}$$

can be established. Their dynamics consist of n_y integrator chains, each of which has $p_{i_y} - 1$ integrators and one input-dependent function. The system output y_{i_y} is the output of the last integrator of the chain:

$$\dot{\boldsymbol{\xi}}_i = \begin{bmatrix} \dot{\xi}_{i,1} \\ \vdots \\ \dot{\xi}_{i,p_i - 1} \\ \dot{\xi}_{i,p_i} \end{bmatrix} = \begin{bmatrix} \xi_{i,2} \\ \vdots \\ \xi_{i,p_i} \\ w_i(\boldsymbol{x}) + \boldsymbol{v}_i(\boldsymbol{x}) \boldsymbol{u} \end{bmatrix} , \quad y = \xi_{i,1} \, . \tag{2.41}$$

The row vector $\boldsymbol{v}_i$ is the ith row of the matrix $\underline{\boldsymbol{V}}$ from (2.35) and w_i is the ith element of the vector

$$\boldsymbol{w}(\boldsymbol{x}) = \begin{bmatrix} L_{\boldsymbol{f}}^{p_1} h_1(\boldsymbol{x}) \\ \vdots \\ L_{\boldsymbol{f}}^{p_{n_y}} h_{n_y}(\boldsymbol{x}), \end{bmatrix} . \tag{2.42}$$

Exact tracking of a desired output $\boldsymbol{y}^*(t)$ can be achieved with the p_ith derivative of the ith output trajectory $\overset{(p_i)}{y_i^*}$ as

$$\boldsymbol{u} = \underline{\boldsymbol{V}}^{-1}(\boldsymbol{x}(t)) \left(-\boldsymbol{w}(\boldsymbol{x}(t)) + \begin{bmatrix} \overset{(p_1)}{y_1^*} \\ \vdots \\ \overset{(p_{n_y})}{y_{n_y}^*} \end{bmatrix} \right) . \tag{2.43}$$

Variables with a star $(\cdot)^*$ note a desired value, also called trajectory. By adding error terms of the first to $(p_i - 1)$th derivative of the trajectory, asymptotic tracking can be achieved with the control law

$$\begin{aligned} \boldsymbol{u} = \underline{\boldsymbol{V}}^{-1}(\boldsymbol{x}(t)) \Bigg(-\boldsymbol{w}(\boldsymbol{x}(t)) + \begin{bmatrix} \overset{(p_1)}{y_1^*} \\ \vdots \\ \overset{(p_{n_y})}{y_{n_y}^*} \end{bmatrix} \\ - \begin{bmatrix} \sum_{i_1=1}^{p_1} k_{1,i_1-1} \left(L_{\boldsymbol{f}}^{i_1-1} h_1(\boldsymbol{x}) - \overset{(i_1-1)}{y_1^*} \right) \\ \vdots \\ \sum_{i_{n_y}=1}^{p_{n_y}} k_{n_y,i_{n_y}-1} \left(L_{\boldsymbol{f}}^{i_{n_y}-1} h_{n_y}(\boldsymbol{x}) - \overset{(i_{n_y}-1)}{y_1^*} \right) \end{bmatrix} \Bigg) . \end{aligned} \tag{2.44}$$

The controller gains of the linearized and decoupled closed loop system are

$$\boldsymbol{k}_j^\top = [k_{j,0}, \ldots, k_{j,p_j-1}] \quad \text{with } j = 1 \ldots n_y \tag{2.45}$$

and the actual values of the output derivatives are expressed in terms of the states with Lie derivatives.

If the sum of elements of the vector relative degree $\boldsymbol{p}$ from Lemma 1 is less than the system order, $\sum_{i=1}^{n_y} p_i = p < n$, *internal* dynamics result, which cannot be seen from the outputs $\boldsymbol{y}$. The states of these internal dynamics are $n - p$ additional coordinates

$$\boldsymbol{\eta} = \begin{bmatrix} \gamma_{p+1} \\ \vdots \\ \gamma_n \end{bmatrix} , \tag{2.46}$$

such that the full mapping of the original states $\boldsymbol{x}$ to the new coordinates results as

$$\begin{bmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_n \end{bmatrix} = \boldsymbol{\gamma}(\boldsymbol{x}) . \tag{2.47}$$

In general, the internal dynamics are nonlinear and input-affine, i.e. they can be expressed as

$$\dot{\boldsymbol{\eta}} = \boldsymbol{f}_\eta(\boldsymbol{\xi}, \boldsymbol{\eta}) + \underline{\boldsymbol{g}}_\eta(\boldsymbol{\xi}, \boldsymbol{\eta}) \boldsymbol{u} . \tag{2.48}$$

Due to the non-observability of $\boldsymbol{\eta}$, these dynamics (2.48) cannot be stabilized with an additional output feedback controller. The stability of (2.48) must therefore be investigated additionally to the design of a stabilizing controller. This investigation is usually done for the equilibrium $\boldsymbol{y} = \boldsymbol{0}$ without loss of generality, since any equilibrium can be shifted to zero. The concept of *zeroing all outputs* is applied, which yields the coordinates $\boldsymbol{\xi}$ being and remaining zero [Isi95, p. 225]

$$\boldsymbol{\xi}(t) = \boldsymbol{0} \quad \forall t \; . \tag{2.49}$$

To ensure (2.49), the input must be chosen as

$$\boldsymbol{u}(t) = -\underline{\boldsymbol{V}}^{-1} \left(\boldsymbol{\gamma}^{-1}(\boldsymbol{\xi} = \boldsymbol{0}, \boldsymbol{\eta}(t))\right) \boldsymbol{w} \left(\boldsymbol{\gamma}^{-1}(\boldsymbol{\xi} = \boldsymbol{0}, \boldsymbol{\eta}(t))\right) \tag{2.50}$$

with the inverse coordinate mapping $\boldsymbol{\gamma}^{-1}$, that results by inverting (2.47). Using (2.49) and (2.50) in the dynamics (2.48), the *zero dynamics* are obtained and can be analyzed to assess the stability of the internal dynamics. If this stability can be proven, the stability of the full closed-loop system, with the asymptotically stabilizing feedback law (2.44), can be concluded.

2.5 Stability of Nonlinear Systems

A stability investigation of the nonlinear autonomous system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{2.51}$$

is less straightforward than for linear systems. Two methods are briefly presented in this context, since they will be applied later in the controller synthesis chapter: Lyapunov's direct and indirect method. The description follows closely the one in KHALIL [Kha02], where further details about these and other methods can be found. Both approaches assume an equilibrium point of the system (2.51) at $\boldsymbol{x}_0 = \boldsymbol{0}$, which means that

$$\dot{\boldsymbol{x}}_0 = \boldsymbol{f}(\boldsymbol{x}_0) = \boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0} \; . \tag{2.52}$$

As in Section 2.4, this does not imply a loss of generality, since an equilibrium at coordinates other than $\boldsymbol{x} = \boldsymbol{0}$ can always be shifted to zero.

An analysis of the full nonlinear system can be performed with *Lyapunov's direct method* if a Lyapunov function can be found. A continuously differentiable Lyapunov function candidate $V(\boldsymbol{x}) : \mathcal{D} \to \mathbb{R}$, with the domain $\mathcal{D} \subset \mathbb{R}^n$ containing the equilibrium point $\boldsymbol{x}_0 = \boldsymbol{0}$ of (2.51), and the properties

$$V(\boldsymbol{x} = \boldsymbol{0}) = 0 \quad \text{and} \quad V(\boldsymbol{x}) > 0 \quad \forall \boldsymbol{x} \in \mathcal{D} \setminus \{\boldsymbol{0}\} \tag{2.53}$$

is searched. Local stability can in some cases be concluded from the properties of the derivative of V:

- $\dot{V}(\boldsymbol{x}) \leq 0$ (negative semidefinite): The system (2.51) is locally stable,
- $\dot{V}(\boldsymbol{x}) < 0$ (negative definite): The system (2.51) is locally asymptotically stable,
- $\dot{V}(\boldsymbol{x}) \not\leq 0$: No conclusion about the stability of the system (2.51) can be drawn.

The Lyapunov function candidate (2.53) is called a Lyapunov function if its derivative is negative definite or negative semidefinite, i.e. one of the first two variants from above. While this stability proof is straightforward when a Lyapunov function $V(\boldsymbol{x})$ is available, finding it is generally not easy.

If such a Lyapunov function cannot be found, *Lyapunov's indirect method* can be used to assess the stability of the nonlinear system for a small region around the equilibrium point. This method uses the linearization of (2.51) around the equilibrium point at $\boldsymbol{x}_0 = \boldsymbol{0}$,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{f}(\boldsymbol{x}_0) + \underbrace{\left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_0=\boldsymbol{0}}}_{\underline{\boldsymbol{A}}_{\text{lin}}} (\boldsymbol{x} - \boldsymbol{x}_0) = \underline{\boldsymbol{A}}_{\text{lin}} \boldsymbol{x} \ . \tag{2.54}$$

The real parts of $\underline{\boldsymbol{A}}_{\text{lin}}$'s eigenvalues λ_i are analyzed for the stability investigation:

- $\text{Re}(\lambda_i) < 0 \ \forall i$: The equilibrium is asymptotically stable,
- $\text{Re}(\lambda_i) > 0$ for one or more i: The equilibrium is unstable,
- $\text{Re}(\lambda_i) \leq 0 \ \forall i$: No conclusion about the stability of the equilibrium can be drawn.

2.6 State and Output Controllability

Controllability is a property of dynamical systems that describes whether it is possible to reach arbitrary system states or outputs by appropriate inputs. Therefore, it is always important to check if a system is controllable before designing a controller for it. In this section, two different types of controllability are defined and criteria to check linear and nonlinear systems for controllability are revisited. The derivations follow the books of Lunze [Lun14] for linear and Adamy [Ada14] for nonlinear systems.

First, let us define complete state controllability, which is oftentimes just called controllability, since it is the most common form.

Definition 1 (Complete state controllability) *A system is called completely state controllable, if it can be transferred from an initial state $\boldsymbol{x}_0$ to any state $\boldsymbol{x}(t_\text{e})$ in finite time by an appropriate choice of the system inputs $\boldsymbol{u}$.*

For a linear MIMO system, the controllability can e.g. be checked with Kalman's criterion.

Lemma 2 *A linear system*

$$\dot{\boldsymbol{x}} = \underline{\boldsymbol{A}}\boldsymbol{x} + \underline{\boldsymbol{B}}\boldsymbol{u} \tag{2.55}$$

of order n_x, i.e. with n_x states, is fully state controllable if and only if the rank of the state controllability matrix

$$\underline{\boldsymbol{S}}_{\mathrm{sc}} = \begin{bmatrix} \underline{\boldsymbol{B}} & \underline{\boldsymbol{A}}\underline{\boldsymbol{B}} & \underline{\boldsymbol{A}}^2\underline{\boldsymbol{B}} & \dots & \underline{\boldsymbol{A}}^{n-1}\underline{\boldsymbol{B}} \end{bmatrix} \tag{2.56}$$

is equal to the system order n_x [Lun14].

Such a controllability criterion does not exist for the general nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \, . \tag{2.57}$$

However, assume that a control-affine system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \underline{\boldsymbol{g}}(\boldsymbol{x})\boldsymbol{u} \tag{2.58}$$

with n states and m inputs is represented in the nonlinear canonical form

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_{n_1-1} \\ \dot{x}_{n_1} \\ \dot{x}_{n_1+1} \\ \vdots \\ \dot{x}_{n_2} \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} x_2 \\ \vdots \\ x_{n_1} \\ a_1(\boldsymbol{x}) \\ x_{n_1+2} \\ \vdots \\ a_2(\boldsymbol{x}) \\ \vdots \\ a_m(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ b_{1,1}(\boldsymbol{x}) & b_{1,2}(\boldsymbol{x}) & \dots & b_{1,m}(\boldsymbol{x}) \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & b_{2,2}(\boldsymbol{x}) & \dots & b_{2,m}(\boldsymbol{x}) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_{m,m}(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} . \tag{2.59}$$

Then, the controllability can be checked with the following criterion.

Lemma 3 *A nonlinear, control-affine system* (2.58) *with $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{u} \in \mathbb{R}^m$ that is described in the nonlinear canonical form* (2.59)*, or can be transformed into this form with an unambiguous and continuously differentiable coordinate transformation, is controllable if*

$$b_{i,i} \neq 0 \quad \forall \, i = 1 \dots m \, . \tag{2.60}$$

Output controllability is a less common form of the controllability and it is defined analogously to the state controllability

Definition 2 (Complete output controllability) *A system is called completely output controllable, if its output can be transferred from an initial value $\boldsymbol{y}_0$ to any value $\boldsymbol{y}(t_{\mathrm{e}})$ in finite time by an appropriate choice of the system inputs $\boldsymbol{u}$.*

For linear systems, KALMAN's criterion from Lemma 2 can be extended for output controllability.

Lemma 4 *A linear system* $(\underline{\boldsymbol{A}}, \underline{\boldsymbol{B}}, \underline{\boldsymbol{C}}, \underline{\boldsymbol{D}})$ *is completely output controllable if and only if the rank of the output controllability matrix*

$$\underline{\boldsymbol{S}}_{\mathrm{oc}} = \begin{bmatrix} \underline{\boldsymbol{C}}\,\underline{\boldsymbol{B}} & \underline{\boldsymbol{C}}\,\underline{\boldsymbol{A}}\,\underline{\boldsymbol{B}} & \underline{\boldsymbol{C}}\,\underline{\boldsymbol{A}}^2\,\underline{\boldsymbol{B}} & \dots & \underline{\boldsymbol{C}}\,\underline{\boldsymbol{A}}^{n-1}\,\underline{\boldsymbol{B}} & \underline{\boldsymbol{D}} \end{bmatrix} \tag{2.61}$$

is equal to the number of outputs [Lun14].

For the state controllable linear system $(\underline{\boldsymbol{A}}, \underline{\boldsymbol{B}}, \underline{\boldsymbol{C}}, \underline{\boldsymbol{D}})$ output controllability can easily be checked. The outputs must simply be linearly independent of each other and dependent on the states, since the latter are controllable. This can be checked with the rank of the output matrix, which must equal the number of outputs, i.e.

$$\mathrm{rank}(\underline{\boldsymbol{C}}) = n_y \,. \tag{2.62}$$

Finally, note that the above definitions only concern *complete* (output) controllability. This means that any state / output within the full state / output space can be reached. If a system cannot be proven completely controllable, it might still be controllable in a sub-space of the full state / output space.

2.7 Control Allocation

A system that has more manipulated variables $\boldsymbol{u}$ than control variables $\boldsymbol{y}$, assuming controllability, leads to infinitely many solutions $\boldsymbol{u}^*$ to achieve the desired system output $\boldsymbol{y}^*$. The associated control problem of finding a solution to achieve $\boldsymbol{y}^*$ in such a system is called *control allocation*.

This abstract definition can be specified for mechatronic systems, like the considered rover, marine vessels or aircrafts, see [JF13] for specific examples. Actuator or other effector inputs represent the manipulated variables $\boldsymbol{u}$ and control the behavior of the mechatronic system, which is represented in the system outputs $\boldsymbol{y}$. The latter are oftentimes velocities of the system and the system is said to be *over-actuated*, since it has more actuators, i.e. manipulated variables, than it would need to achieve the desired motion $\boldsymbol{y}^*$.

In general, the relation between inputs $\boldsymbol{u}$ and outputs $\boldsymbol{y}$ of a system is dynamic, mostly described by ordinary differential equations. For finding the manipulated variable trajectories $\boldsymbol{u}^*$, it can be beneficial to separate the full over-actuated system dynamics into three subsystems, as proposed by Johansen and Fossen [JF13]. This yields less complex systems that can be handled more easily with common control methods, compared to the full system. The three subsystems are depicted in the green box of Figure 1.4:

- System dynamics: Virtual inputs are chosen such that the system level dynamics have the same number of in- and outputs and are output controllable. These virtual inputs are usually forces and torques for a mechatronic system and they are called virtual since they are, generally, not directly controlled by actuators.

- Static allocation: This subsystem relates the virtual inputs to the actual effector forces and torques, which basically is a force and torque sum that can be dependent e.g. on varying kinematics of the system. In the linear case, it is written in the form of an underdetermined system of equations (1.1).
- Effector dynamics: The actuators or effectors of the system might require another model to adequately represent their individual dynamic behavior. Actuator inputs, such as motor currents are therein linked to forces or torques that are needed for the allocation. Like the system dynamics, the effector dynamics have the same number of in- and outputs and must be output controllable.

The advantage of this separation is that the allocation itself can be performed on an, oftentimes linear, static system of equations. No additional constraints are introduced and thus all DoFs are kept available for optimization.

For completeness, the underdetermined linear system of equations (1.1)

$$\underline{\boldsymbol{A}}\boldsymbol{x} = \boldsymbol{b} \tag{2.63}$$

is repeated here from Section 1.3 with the matrix $\underline{\boldsymbol{A}} \in \mathbb{R}^{(n_b \times n_x)}$ having more columns n_x than rows n_b and thus being generally non-invertible. As explained in Section 1.3, (2.63) has infinitely many solutions if $\underline{\boldsymbol{A}}$ has full row rank and its general solution is

$$\boldsymbol{x} = \underline{\boldsymbol{A}}^{+}\boldsymbol{b} + \underline{\boldsymbol{A}}^{\perp}\boldsymbol{\rho} \ . \tag{2.64}$$

The first term features the pseudo-inverse $\underline{\boldsymbol{A}}^{+}$ of the $\underline{\boldsymbol{A}}$-matrix and represents the least-squares solution of (2.63). The second term, the homogeneous solution of (2.63), allows to adjust the solution with the DoFs $\boldsymbol{\rho}$ and a base of the $\underline{\boldsymbol{A}}$-matrix's null-space $\underline{\boldsymbol{A}}^{\perp}$. For an over-actuated system as described above, the number of DoFs is the difference between the number of manipulation and control variables

$$n_\rho = n_x - n_b \ . \tag{2.65}$$

In robotics, over-actuation is commonly referred to as kinematic redundancy and the number n_ρ from (2.65) is called the *degree of redundancy* [Sic+10].

3 Model Development

For a mobility controller of a wheeled mobile robot, only its locomotion subsystem is important. Components, such as scientific instruments or cameras, that do not move relatively to the main body are not considered here. A locomotion subsystem generally consists of a main body, several wheels, motors to drive and potentially steer and the suspension. The latter can consist of fixed parts as well as active and passive joints. The specific locomotion subsystem of the considered ExoMars BB2 rover is described in Section 3.1. Additionally to the parts of the robot, the contact between wheels and ground plays a crucial role, especially on soft terrain.

This chapter shows the development of models for the whole locomotion subsystem. It is started with the rover kinematics and differential kinematics in Section 3.2 and the dynamics, with the assumption of one rigid body, in Section 3.3. The dynamics of each individual wheel, consisting of steering and drive actuation, are modeled in Section 3.4. Existing terramechanics models are revisited in Section 3.5, before a novel empirical model for time critical simulations is developed in Section 3.6. Due to limitations for the usage of models in the controller synthesis, a simplified contact model is derived in Section 3.7 and combined with the wheel dynamics to one nonlinear effector model in Section 3.8. An overview of the developed sub-models and their interfaces is finally provided in Section 3.9.

3.1 Rosalind Franklin and ExoMars BB2 Rover

The rover of ESA's ExoMars mission was recently renamed Rosalind Franklin rover and is pictured in Figure 3.1. To test the locomotion capabilities, the ExoMars BB2 Rover was built in earlier phases of the mission and is used within this work. As opposed to the full rover, the BB2 has one main body without a mast, pan-tilt or drilling unit. However, its locomotion system is representative of the final rover, see Figure 3.2. It consists of three *bogies*, which are mounted on the left, right and rear sides of the main body. Each of them holds two wheel units, consisting of a *deployment*, *steering* and *drive* actuation, driven with a brushed DC motor and different gear combinations. The wheels are flexible through spring steel strips with the goal to achieve a larger contact patch for better traction and climbing abilities over obstacles.

It is planned within the ExoMars mission, to only use the deployment actuators for initial unfolding of the rover, not for driving. However, if they would be used, a special locomotion mode, called wheel-walking, would be possible [Wie17]. For the present work, the deployment actuators are neglected, since this is the baseline for the ExoMars operations and other rovers are not generally equipped with such actuators. Furthermore, the wheel deformation

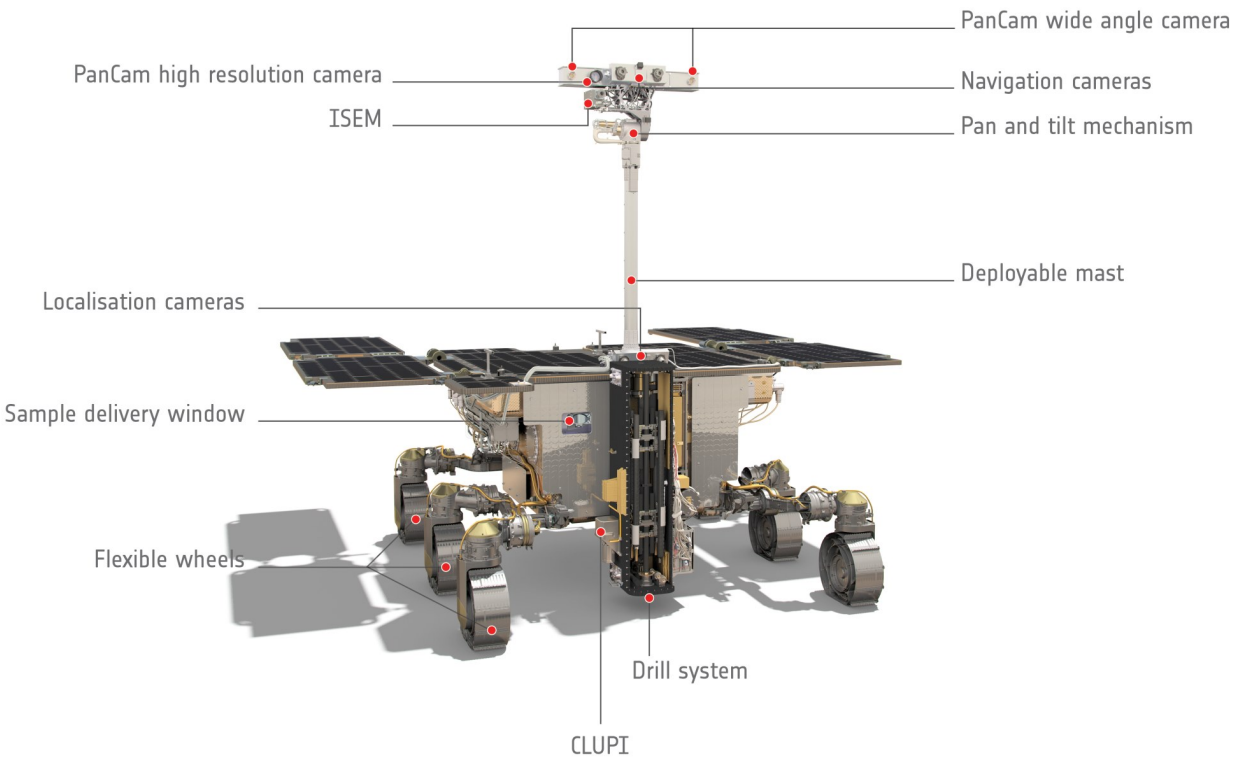


Figure 3.1: The Rosalind Franklin / ExoMars rover with its main parts; © ESA/ATG medialab.

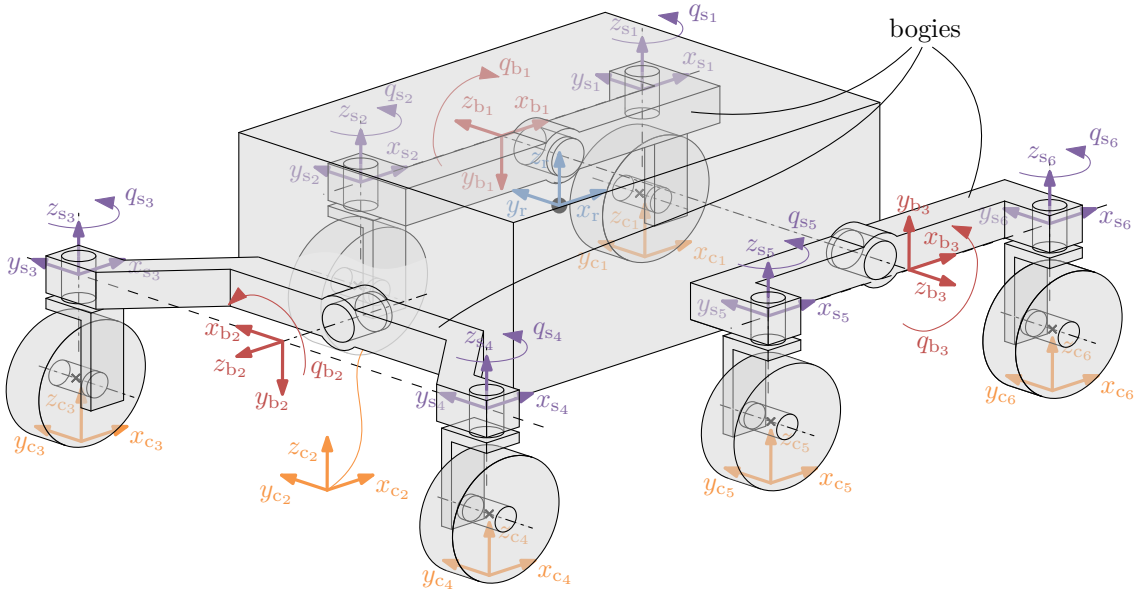


Figure 3.2: Architecture of the ExoMars BB2 rover.

Table 3.1: DH parameters of the ExoMars BB2 rover for its parallel kinematic chains from CoG to the six contact points.

Transformation	ϑ	d	a	α
$\mathrm{r} \to \mathrm{b}_1$	0	$r_{\mathrm{b}_1\mid\mathrm{r},z}$	$r_{\mathrm{b}_1\mid\mathrm{r},x}$	$-\pi/2$
	0	$r_{\mathrm{b}_1\mid\mathrm{r},y}$	0	0
$\mathrm{r} \to \mathrm{b}_2$	$\pi/2$	$r_{\mathrm{b}_2\mid\mathrm{r},z}$	$-r_{\mathrm{b}_2\mid\mathrm{r},y}$	$-\pi/2$
	0	$-r_{\mathrm{b}_2\mid\mathrm{r},x}$	0	0
$\mathrm{r} \to \mathrm{b}_3$	0	$r_{\mathrm{b}_3\mid\mathrm{r},z}$	$r_{\mathrm{b}_3\mid\mathrm{r},x}$	$\pi/2$
	0	$-r_{\mathrm{b}_3\mid\mathrm{r},y}$	0	0
$\mathrm{b}_1 \to \mathrm{s}_i$	$q_{\mathrm{b},1}$	$r_{\mathrm{s}_i\mid\mathrm{b}_1,y}$	$r_{\mathrm{s}_i\mid\mathrm{b}_1,x}$	$\pi/2$
for $i = \{1, 2\}$	0	$r_{\mathrm{s}_i\mid\mathrm{b}_1,z}$	0	0
$\mathrm{b}_2 \to \mathrm{s}_i$	$q_{\mathrm{b},2}$	$-r_{\mathrm{s}_i\mid\mathrm{b}_2,x}$	$r_{\mathrm{s}_i\mid\mathrm{b}_2,y}$	$\pi/2$
for $i = \{3, 4\}$	$-\pi/2$	$r_{\mathrm{s}_i\mid\mathrm{b}_2,z}$	0	0
$\mathrm{b}_3 \to \mathrm{s}_i$	$q_{\mathrm{b},3}$	$r_{\mathrm{s}_i\mid\mathrm{b}_3,y}$	$r_{\mathrm{s}_i\mid\mathrm{b}_3,x}$	$\pi/2$
for $i = \{5, 6\}$	0	$r_{\mathrm{s}_i\mid\mathrm{b}_3,z}$	0	0
$\mathrm{s}_i \to \mathrm{c}_i$	q_{s_i}	$r_{\mathrm{c}_i\mid\mathrm{s}_i,z}$	0	0
for $i = 1 \ldots 6$				

on soft soil is very small and the flexibility of the wheels is thus neglected. For the following sections, different coordinate systems are used to set up the model equations, see Figure 3.2. The rover's Center of Gravity (CoG) is in the 'r'-frame and each of the three bogies has a 'b_j'-frame. On each steering axis is a 's_i'-frame and six contact frames 'c_i' are located at the assumed wheel-ground contact points. The orientation of the bogie and steering frames is such that the rotation axes are in the z-axis of the respective system, details are described in Section 3.2. Similar to the convention used for road vehicles, rover and contact frames are oriented with their z-axis pointing up, x forward and y sideways.

3.2 Rover Kinematics and Differential Kinematics

In this section the kinematics and differential kinematics of the ExoMars BB2 rover are derived following the DH convention that was introduced in Section 2.2.5. More details on this approach can be found in the publications of an earlier version of the chassis controller [BZ17; BB18]. With the joint coordinates for the bogies and steering joints, $\boldsymbol{q}_\mathrm{b}$ and $\boldsymbol{q}_\mathrm{s}$, the total vector of joint coordinates reads

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_\mathrm{b} \\ \boldsymbol{q}_\mathrm{s} \end{bmatrix} \in \mathbb{R}^{(9\times 1)} \,. \tag{3.1}$$

The rover kinematics are described as six parallel kinematic chains starting at the 'r'-frame and going, through the 'b'- and 's'-frames, to the wheel-ground contact 'c'-frames. To cope with translations in all three dimensions with the DH systematic, two sets of DH parameters

per transformation are needed for most of them. The resulting DH parameter table is shown in Table 3.1. One homogeneous transformation matrix is obtained for each transformation through (2.22) and the full kinematic chains are derived by matrix multiplications through (2.19) as

$$\begin{aligned}
&\text{left front:} && {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{c}_1} = {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{b}_1}\,{}^{\mathrm{b}_1}\underline{\boldsymbol{T}}_{\mathrm{s}_1}\,{}^{\mathrm{s}_1}\underline{\boldsymbol{T}}_{\mathrm{c}_1}\,,\\
&\text{left middle:} && {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{c}_2} = {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{b}_1}\,{}^{\mathrm{b}_1}\underline{\boldsymbol{T}}_{\mathrm{s}_2}\,{}^{\mathrm{s}_2}\underline{\boldsymbol{T}}_{\mathrm{c}_2}\,,\\
&\text{left rear:} && {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{c}_3} = {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{b}_2}\,{}^{\mathrm{b}_2}\underline{\boldsymbol{T}}_{\mathrm{s}_3}\,{}^{\mathrm{s}_3}\underline{\boldsymbol{T}}_{\mathrm{c}_3}\,,\\
&\text{right rear:} && {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{c}_4} = {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{b}_2}\,{}^{\mathrm{b}_2}\underline{\boldsymbol{T}}_{\mathrm{s}_4}\,{}^{\mathrm{s}_4}\underline{\boldsymbol{T}}_{\mathrm{c}_4}\,,\\
&\text{right middle:} && {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{c}_5} = {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{b}_3}\,{}^{\mathrm{b}_3}\underline{\boldsymbol{T}}_{\mathrm{s}_5}\,{}^{\mathrm{s}_5}\underline{\boldsymbol{T}}_{\mathrm{c}_5}\,,\\
&\text{right front:} && {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{c}_6} = {}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{b}_3}\,{}^{\mathrm{b}_3}\underline{\boldsymbol{T}}_{\mathrm{s}_6}\,{}^{\mathrm{s}_6}\underline{\boldsymbol{T}}_{\mathrm{c}_6}\,.
\end{aligned} \tag{3.2}$$

Due to the assumption of zero contact angles, see Section 1.5, the ith contact point is always on the ith steering axis. The $\boldsymbol{r}_{\mathrm{c}_i|\mathrm{r}}$-vector, which is contained in the homogeneous transformation matrix ${}^{\mathrm{r}}\underline{\boldsymbol{T}}_{\mathrm{c}_i}$, is in this case only dependent on the bogie angles $\boldsymbol{q}_{\mathrm{b}}$, not the steering angles $\boldsymbol{q}_{\mathrm{s}}$. Following (2.28), the relative velocity of the ith contact point can be computed as

$$\boldsymbol{v}_{\mathrm{c}_i|\mathrm{r}} = \underline{\boldsymbol{J}}_{i,\mathrm{rel}}(\boldsymbol{q}_{\mathrm{b}})\dot{\boldsymbol{q}}_{\mathrm{b}} \tag{3.3}$$

with the ith velocity Jacobian $\underline{\boldsymbol{J}}_{i,\mathrm{rel}} \in \mathbb{R}^{(3\times 3)}$. The index 'rel' is added to the Jacobian here since it only yields the contact point velocity $\boldsymbol{v}_{\mathrm{c}_i|\mathrm{r}}$ relative to the rover CoG frame 'r'.

The 'r'-frame is, however, not fixed in space but translates and rotates, with respect to the inertial '0'-frame, with the velocities $\boldsymbol{v}_{\mathrm{r}|0}$ and $\boldsymbol{\omega}_{\mathrm{r}|0}$. To get to the total wheel velocity of the ith contact point, see (2.24), the translational rover velocity needs to be added to (3.3) and, additionally, the cross product of the 'r'-frame rotation and its distance to the ith contact frame needs to be taken into account. With the anticommutativity of the cross product operation, $\boldsymbol{\omega} \times \boldsymbol{r} = -\boldsymbol{r} \times \boldsymbol{\omega}$, and the cross-product building matrix $\underline{\boldsymbol{S}}$ from (2.26), the absolute contact point velocity follows as

$$\begin{aligned}
{}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{c}_i|0} &= {}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r}|0} + {}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r}|0} \times \boldsymbol{r}_{\mathrm{c}_i|\mathrm{r}} + \boldsymbol{v}_{\mathrm{c}_i|\mathrm{r}}\\
&= {}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r}|0} - \boldsymbol{r}_{\mathrm{c}_i|\mathrm{r}} \times {}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r}|0} + \boldsymbol{v}_{\mathrm{c}_i|\mathrm{r}}\\
&= \underbrace{\begin{bmatrix}\underline{\boldsymbol{I}} & -\underline{\boldsymbol{S}}(\boldsymbol{r}_{\mathrm{c}_i|\mathrm{r}}(\boldsymbol{q}_{\mathrm{b}})) & \underline{\boldsymbol{J}}_{i,\mathrm{rel}}(\boldsymbol{q}_{\mathrm{b}})\end{bmatrix}}_{=\underline{\boldsymbol{J}}_{i,\mathrm{abs}}(\boldsymbol{q}_{\mathrm{b}})} \underbrace{\begin{bmatrix}{}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r}|0}\\ {}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r}|0}\\ \dot{\boldsymbol{q}}_{\mathrm{b}}\end{bmatrix}}_{\dot{\tilde{\boldsymbol{q}}}}\,.
\end{aligned} \tag{3.4}$$

With the rotation matrix from frame 'r' to 's_i', the contact point velocities ${}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{c}_i|0}$ can be expressed in the non-steered 's_i'-frame

$${}^{\mathrm{s}_i}\boldsymbol{v}_{\mathrm{c}_i|0} = {}^{\mathrm{s}_i}\underline{\boldsymbol{R}}_{\mathrm{r}}(\boldsymbol{q}_{\mathrm{b}})\,{}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{c}_i|0} = \underbrace{{}^{\mathrm{s}_i}\underline{\boldsymbol{R}}_{\mathrm{r}}(\boldsymbol{q}_{\mathrm{b}})\underline{\boldsymbol{J}}_{i,\mathrm{abs}}(\boldsymbol{q}_{\mathrm{b}})}_{=\underline{\boldsymbol{J}}_{\mathrm{c}_i}(\boldsymbol{q}_{\mathrm{b}})}\dot{\tilde{\boldsymbol{q}}}\,. \tag{3.5}$$

The angles of the steering joints $\boldsymbol{q}_{\mathrm{s}}$ and their derivatives $\dot{\boldsymbol{q}}_{\mathrm{s}}$ thus have no influence on the contact point velocities as long as the latter are expressed in their respective 's_i'-frame. Only

if the velocities are rotated once more to express them in the 'c_i'-frame, a dependency on the steering joint angles $\boldsymbol{q}_\mathrm{s}$ arises:

$$ {}^{\mathrm{c}_i}\boldsymbol{v}_{\mathrm{c}_i|0} = {}^{\mathrm{c}_i}\underline{\boldsymbol{R}}_{\mathrm{s}_i}(\boldsymbol{q}_\mathrm{s})\,{}^{\mathrm{s}_i}\boldsymbol{v}_{\mathrm{c}_i|0} = {}^{\mathrm{c}_i}\underline{\boldsymbol{R}}_{\mathrm{s}_i}(\boldsymbol{q}_\mathrm{s})\underline{\boldsymbol{J}}_{\mathrm{c}_i}(\boldsymbol{q}_\mathrm{b})\dot{\tilde{\boldsymbol{q}}} \,. \tag{3.6} $$

Through the kineto-statics duality (2.31), the transformation of the ith contact point forces $\boldsymbol{F}_{\mathrm{c}_i}$ to the rover, as well as the bogie joint torques $\boldsymbol{\tau}_{q_\mathrm{b}}$, can be computed with the transposed Jacobian as

$$ \begin{bmatrix} \boldsymbol{F}_{\mathrm{r},i} \\ \boldsymbol{T}_{\mathrm{r},i} \\ \boldsymbol{\tau}_{q_\mathrm{b},i} \end{bmatrix} = \left({}^{\mathrm{c}_i}\underline{\boldsymbol{R}}_{\mathrm{s}_i}(\boldsymbol{q}_\mathrm{s})\underline{\boldsymbol{J}}_{\mathrm{c}_i}(\boldsymbol{q}_\mathrm{b})\right)^\top \boldsymbol{F}_{\mathrm{c}_i} = \underline{\boldsymbol{J}}_{\mathrm{c}_i}^\top(\boldsymbol{q}_\mathrm{b})\,{}^{\mathrm{s}_i}\boldsymbol{F}_{\mathrm{c}_i} \,. \tag{3.7} $$

Note that the Jacobian in (3.6) and its transpose in (3.7) is only the translational part compared to the full geometric Jacobian from (2.30). This is valid for the force and torque transformation (3.7) due to the assumption of no torque transfer in the contact point, $\boldsymbol{T}_{\mathrm{c}_i} = \boldsymbol{0}$. It can be seen that (3.7) is derived for the contact forces expressed in the non-steered 's_i'-frame instead of the contact frame. This eliminates the steering angles $\boldsymbol{q}_\mathrm{s}$ from the equation, which allows to only consider them in the underlying wheel-ground sub-controller instead of in the allocation. This is beneficial for the desired modular control approach.

Summing up all $n = 6$ contact force vectors, the total static force-torque equilibrium is derived as

$$ \begin{bmatrix} \boldsymbol{F}_\mathrm{r} \\ \boldsymbol{T}_\mathrm{r} \\ \boldsymbol{\tau}_{q_\mathrm{b}} \end{bmatrix} = \underbrace{\begin{bmatrix} \underline{\boldsymbol{J}}_{\mathrm{c}1}^\top(\boldsymbol{q}_\mathrm{b}) & \dots & \underline{\boldsymbol{J}}_{\mathrm{c}6}^\top(\boldsymbol{q}_\mathrm{b}) \end{bmatrix}}_{\underline{\boldsymbol{J}}^\top(\boldsymbol{q}_\mathrm{b})} \underbrace{\begin{bmatrix} {}^{\mathrm{s}1}\boldsymbol{F}_{\mathrm{c}1} \\ \vdots \\ {}^{\mathrm{s}6}\boldsymbol{F}_{\mathrm{c}6} \end{bmatrix}}_{{}^{\mathrm{s}}\boldsymbol{F}_\mathrm{c}} \,. \tag{3.8} $$

Due to the friction-less rotating bogie joints, they cannot transfer torque, $\boldsymbol{\tau}_{q_\mathrm{b}} = \boldsymbol{0}$, and only the rover force and torque rows of (3.8) are of interest in the remainder:

$$ \underbrace{\begin{bmatrix} \boldsymbol{F}_\mathrm{r} \\ \boldsymbol{T}_\mathrm{r} \end{bmatrix}}_{(6\times1)} = \underbrace{\underline{\boldsymbol{J}}_F^\top(\boldsymbol{q}_\mathrm{b})}_{(6\times18)} \underbrace{{}^{\mathrm{s}}\boldsymbol{F}_\mathrm{c}}_{(18\times1)} \,. \tag{3.9} $$

Constraining equations for the contact forces can be obtained by setting the $\boldsymbol{\tau}_{q_\mathrm{b}}$-rows of (3.8) to zero, which could be used for the quasi-static normal force estimation. However, further simplifications for the force directions will be made in Section 3.7.2.

3.3 Rigid Body Dynamics

The rigid body dynamics sub-model relates the rover movement to the total rover forces, which are needed for the force-torque balance (3.9). Due to the moderate terrain, see Section 1.5, and the slow maximum velocity of $3\,\frac{\mathrm{cm}}{\mathrm{s}}$, the bogie dynamics are neglected in this work. One mass matrix $\underline{\boldsymbol{M}}_\mathrm{r}$ and one inertia matrix $\underline{\boldsymbol{\Theta}}_\mathrm{r}$ are assumed, combining all parts

of the rover into one rigid body. This body moves and rotates with the velocity ${}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r|0}}$ and angular velocity ${}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r|0}}$ that were introduced in Section 3.2 and are both expressed in the rover frame 'r'. The rover orientation is expressed in the RPY angles, see Section 2.2.2,

$$\boldsymbol{\theta}_{\mathrm{r}} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^{\top} . \tag{3.10}$$

The rigid body is exposed to the earth's gravitational acceleration $\boldsymbol{a}_{\mathrm{g}}$, which is rotated into the 'r'-frame

$${}^{\mathrm{r}}\boldsymbol{a}_{\mathrm{g}} = {}^{\mathrm{r}}\underline{\boldsymbol{R}}_{0}(\boldsymbol{\theta}_{\mathrm{r}})\,{}^{0}\boldsymbol{a}_{\mathrm{g}} = {}^{\mathrm{r}}\underline{\boldsymbol{R}}_{0}(\boldsymbol{\theta}_{\mathrm{r}}) \begin{bmatrix} 0 \\ 0 \\ -9.81\,\frac{\mathrm{m}}{\mathrm{s}^2} \end{bmatrix} . \tag{3.11}$$

External forces and torques exclusively result from the wheel contact forces $\boldsymbol{F}_{\mathrm{c}}$ and were summarized to the total force-torque-vector $[\boldsymbol{F}_{\mathrm{r}}, \boldsymbol{T}_{\mathrm{r}}]^{\top}$ in Section 3.2. Newton's equations of motion thus read

$$\underline{\boldsymbol{M}}_{\mathrm{r}}\,{}^{\mathrm{r}}\dot{\boldsymbol{v}}_{\mathrm{r,total}} = \underline{\boldsymbol{M}}_{\mathrm{r}}\,{}^{\mathrm{r}}\boldsymbol{a}_{\mathrm{g}} + \boldsymbol{F}_{\mathrm{r}} , \tag{3.12}$$

$$\underline{\boldsymbol{\Theta}}_{\mathrm{r}}\,{}^{\mathrm{r}}\dot{\boldsymbol{\omega}}_{\mathrm{r,total}} = \boldsymbol{T}_{\mathrm{r}} . \tag{3.13}$$

To get to the total derivative of the rover's velocity in (3.12), it firstly needs to be transformed to the fixed inertial frame '0',

$$\boldsymbol{v}_{\mathrm{r|0}} = {}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}(\boldsymbol{\theta}_{\mathrm{r}})\,{}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r|0}} , \tag{3.14}$$

such that the total derivative can be computed by applying the chain rule, yielding

$$\begin{aligned} {}^{0}\dot{\boldsymbol{v}}_{\mathrm{r,total}} =& \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{v}_{\mathrm{r|0}} = {}^{0}\dot{\underline{\boldsymbol{R}}}_{\mathrm{r}}(\boldsymbol{\theta}_{\mathrm{r}})\,{}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r|0}} + {}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}(\boldsymbol{\theta}_{\mathrm{r}})\,{}^{\mathrm{r}}\dot{\boldsymbol{v}}_{\mathrm{r|0}} \\ \overset{(2.25)}{=}& \underline{\boldsymbol{S}}(\boldsymbol{\omega}_{\mathrm{r|0}})\,{}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}(\boldsymbol{\theta}_{\mathrm{r}})\,{}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r|0}} + {}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}(\boldsymbol{\theta}_{\mathrm{r}})\,{}^{\mathrm{r}}\dot{\boldsymbol{v}}_{\mathrm{r|0}} . \end{aligned} \tag{3.15}$$

In the derivative of the rotation matrix ${}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}$, the cross-product building $\underline{\boldsymbol{S}}$-matrix was used, see Section 2.3. Equation (3.15) uses the total angular velocity $\boldsymbol{\omega}_{\mathrm{r|0}}$, expressed in the inertial frame. The translational and rotational velocities in (3.15) must, however, be expressed in the same coordinate system to enable a state space representation for the controller synthesis. $\boldsymbol{\omega}_{\mathrm{r|0}}$ is therefore rotated to the 'r'-frame and (2.27) is used to simplify the equation yielding

$$\begin{aligned} {}^{0}\dot{\boldsymbol{v}}_{\mathrm{r,total}} &= \underline{\boldsymbol{S}}({}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}\,{}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r|0}})\,{}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}\,{}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r|0}} + {}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}(\boldsymbol{\theta}_{\mathrm{r}})\,{}^{\mathrm{r}}\dot{\boldsymbol{v}}_{\mathrm{r|0}} \\ &\overset{(2.27)}{=} {}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}\underline{\boldsymbol{S}}({}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r|0}})\,{}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}^{\top}\,{}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}\,{}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r|0}} + {}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}\,{}^{\mathrm{r}}\dot{\boldsymbol{v}}_{\mathrm{r|0}} \\ &= {}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}\underline{\boldsymbol{S}}({}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r|0}})\,{}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r|0}} + {}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}\,{}^{\mathrm{r}}\dot{\boldsymbol{v}}_{\mathrm{r|0}} . \end{aligned} \tag{3.16}$$

Ultimately, (3.16) is rotated back into the rover system to get the desired total CoG linear acceleration, expressed in the 'r'-frame,

$${}^{\mathrm{r}}\dot{\boldsymbol{v}}_{\mathrm{r,total}} = {}^{\mathrm{r}}\underline{\boldsymbol{R}}_{0}\,{}^{0}\dot{\boldsymbol{v}}_{\mathrm{r,total}} = \underline{\boldsymbol{S}}({}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r|0}})\,{}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r|0}} + {}^{\mathrm{r}}\dot{\boldsymbol{v}}_{\mathrm{r|0}} . \tag{3.17}$$

The total angular acceleration, that is used in the angular equation of motion (3.13), is easier to obtain: Applying the above procedure of rotating to the inertial frame (3.14) and

computing the total derivative as in (3.17) yields the cross-product of the rover angular velocity ${}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r}|0}$ with itself, which is zero. This results in the total angular acceleration

$$ {}^{\mathrm{r}}\dot{\boldsymbol{\omega}}_{\mathrm{r,total}} = {}^{\mathrm{r}}\dot{\boldsymbol{\omega}}_{\mathrm{r}|0} \, . \tag{3.18} $$

Using the total linear and angular CoG acceleration expressed in the 'r'-frame from (3.17) and (3.18) and the acceleration (3.11) in the equations of motion (3.12) and (3.13) yields

$$ {}^{\mathrm{r}}\dot{\boldsymbol{v}}_{\mathrm{r}|0} = -\underline{\boldsymbol{S}}({}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r}|0}){}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r}|0} + {}^{\mathrm{r}}\underline{\boldsymbol{R}}_0(\boldsymbol{\theta}_{\mathrm{r}}){}^0\boldsymbol{a}_{\mathrm{g}} + \underline{\boldsymbol{M}}_{\mathrm{r}}^{-1}\boldsymbol{F}_{\mathrm{r}} \, , \tag{3.19} $$

$$ {}^{\mathrm{r}}\dot{\boldsymbol{\omega}}_{\mathrm{r}|0} = \underline{\boldsymbol{\Theta}}_{\mathrm{r}}^{-1}\boldsymbol{T}_{\mathrm{r}} \, . \tag{3.20} $$

A nonlinear state space representation can be obtained with the state vector consisting of the linear and angular velocity as well as the RPY angles,

$$ \boldsymbol{x}_{\mathrm{r}} = \begin{bmatrix} {}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r}|0} \\ {}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r}|0} \\ \boldsymbol{\theta}_{\mathrm{r}} \end{bmatrix} , \tag{3.21} $$

and the inputs being the forces $\boldsymbol{F}_{\mathrm{r}}$ and torques $\boldsymbol{T}_{\mathrm{r}}$. Note that the rover position $\boldsymbol{r}_{\mathrm{r}|0}$ is not part of the state vector, since the dynamics equations (3.19) and (3.20) are not position-dependent. Depending on the normal force model, see Section 3.7.1, the position could be required for the calculation of the contact forces, which influence (3.19) and (3.20) through $\boldsymbol{F}_{\mathrm{r}}$ and $\boldsymbol{T}_{\mathrm{r}}$. This normal force model is, however, not used in the controller synthesis of this work since it would additionally require knowledge of the ground geometry, which is generally not available. For differentiation of the RPY angles $\boldsymbol{\theta}_{\mathrm{r}}$, their definition from Section 2.2.2 is considered. Each of the angles describes a rotation about an axis of the intermediate system that resulted from the last rotation. Therefore, the angular velocities of the individual rotations are only purely around one axis in their respective frame, leading to the total angular velocity

$$ \boldsymbol{\omega}_{\mathrm{r}|0} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \underline{\boldsymbol{R}}_z(\psi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \underline{\boldsymbol{R}}_z(\psi)\underline{\boldsymbol{R}}_y(\theta) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} . \tag{3.22} $$

Rotating the rate vector $\boldsymbol{\omega}_{\mathrm{r}|0}$ into the 'r'-frame with the rotation matrix

$$ {}^{\mathrm{r}}\underline{\boldsymbol{R}}_0(\boldsymbol{\theta}_{\mathrm{r}}) = {}^0\underline{\boldsymbol{R}}_{\mathrm{r}}^{\top}(\boldsymbol{\theta}_{\mathrm{r}}) \overset{(2.14)}{=} \left(\underline{\boldsymbol{R}}_z(\psi)\underline{\boldsymbol{R}}_y(\theta)\underline{\boldsymbol{R}}_x(\phi)\right)^{\top} \tag{3.23} $$

leads to

$$ \begin{aligned} {}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r}|0} &= {}^{\mathrm{r}}\underline{\boldsymbol{R}}_0\boldsymbol{\omega}_{\mathrm{r}|0} \\ &= \underline{\boldsymbol{R}}_x^{\top}(\phi)\underline{\boldsymbol{R}}_y^{\top}(\theta)\underline{\boldsymbol{R}}_z^{\top}(\psi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \underline{\boldsymbol{R}}_x^{\top}(\phi)\underline{\boldsymbol{R}}_y^{\top}(\theta) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \underline{\boldsymbol{R}}_x^{\top}(\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix}}_{=\underline{\boldsymbol{U}}(\boldsymbol{\theta}_{\mathrm{r}})} \underbrace{\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}}_{=\dot{\boldsymbol{\theta}}_{\mathrm{r}}} . \end{aligned} \tag{3.24} $$

$\underline{\boldsymbol{U}}$ is not a rotation matrix, therefore $\underline{\boldsymbol{U}}^{-1} \neq \underline{\boldsymbol{U}}^{\top}$ and the inversion of (3.24) yields

$$\dot{\boldsymbol{\theta}}_{\mathrm{r}} = \underbrace{\frac{1}{\cos(\theta)} \begin{bmatrix} \cos(\theta) & \sin(\phi)\sin(\theta) & \cos(\phi)\sin(\theta) \\ 0 & \cos(\phi)\cos(\theta) & -\sin(\phi)\cos(\theta) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}}_{=\underline{\boldsymbol{U}}^{-1}(\boldsymbol{\theta}_{\mathrm{r}})} {}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r}|0} \, . \tag{3.25}$$

The final rigid body dynamics in nonlinear state space representation is received by combining (3.19), (3.20) and (3.25) to

$$\underbrace{\begin{bmatrix} {}^{\mathrm{r}}\dot{\boldsymbol{v}}_{\mathrm{r}|0} \\ {}^{\mathrm{r}}\dot{\boldsymbol{\omega}}_{\mathrm{r}|0} \\ \dot{\boldsymbol{\theta}}_{\mathrm{r}} \end{bmatrix}}_{\dot{\boldsymbol{x}}_{\mathrm{r}}} = \underbrace{\begin{bmatrix} -\underline{\boldsymbol{S}}({}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r}|0}) {}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r}|0} + {}^{\mathrm{r}}\underline{\boldsymbol{R}}_0(\boldsymbol{\theta}_{\mathrm{r}}) {}^0\boldsymbol{a}_{\mathrm{g}} \\ \boldsymbol{0} \\ \underline{\boldsymbol{U}}^{-1}(\boldsymbol{\theta}_{\mathrm{r}}) {}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r}|0} \end{bmatrix}}_{\boldsymbol{f}_{\mathrm{r}}(\boldsymbol{x}_{\mathrm{r}})} + \underbrace{\begin{bmatrix} \underline{\boldsymbol{M}}_{\mathrm{r}}^{-1} & \underline{\boldsymbol{0}} \\ \underline{\boldsymbol{0}} & \underline{\boldsymbol{\Theta}}_{\mathrm{r}}^{-1} \\ \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} \end{bmatrix}}_{\underline{\boldsymbol{B}}_{\mathrm{r}}} \underbrace{\begin{bmatrix} \boldsymbol{F}_{\mathrm{r}} \\ \boldsymbol{T}_{\mathrm{r}} \end{bmatrix}}_{\boldsymbol{u}_{\mathrm{r}}} . \tag{3.26}$$

3.4 Wheel Dynamics

Although the wheels of the ExoMars BB2 rover are flexible, they are rather stiff compared to the rover weight and the properties of soil. In fact, a deformation was mainly observed for hard ground, driving over single rocks and uphill drives in soft sands with high slippage. Since this work aims at soft soil and uses the assumption of moderately rough terrain with no contact angles, see Section 1.5, the first two cases are not relevant. The high slippage values, for which oscillations and thereby deformation of the wheel were seen, should be avoided for safe rover operations. However, the benefit of modeling the flexible wheels could be investigated in future work.

For the rigid wheel assumption, the angular acceleration of the drive motor is modeled with the input torque $T_{\omega,i}$, the inertia of the gears and wheel in the drive axis $I_{\mathrm{w}_i,y}$, the wheel radius r and the longitudinal contact force $F_{\mathrm{c}_i,x}$, see Figure 3.3,

$$I_{\mathrm{w}_i,y}\dot{\omega}_i = T_{\omega,i} - rF_{\mathrm{c}_i,x} \, . \tag{3.27}$$

The angular acceleration $\ddot{\delta}_i$ of the ith steering is computed with the inertia around the steering axis $I_{\mathrm{w}_i,z}$ and the steering motor torque $T_{\delta,i}$ as

$$I_{\mathrm{w}_i,z}\ddot{\delta}_i = T_{\delta,i} \, . \tag{3.28}$$

These equations of motion are combined to

$$\begin{bmatrix} \dot{\omega}_i \\ \dot{\delta}_i \\ \ddot{\delta}_i \end{bmatrix} = \begin{bmatrix} -\frac{r}{I_{\mathrm{w}_i,y}} F_{\mathrm{c}_i,x} \\ \dot{\delta}_i \\ 0 \end{bmatrix} + \begin{bmatrix} 1/I_{\mathrm{w}_i,y} & 0 \\ 0 & 0 \\ 0 & 1/I_{\mathrm{w}1,z} \end{bmatrix} \begin{bmatrix} T_{\omega,i} \\ T_{\delta,i} \end{bmatrix} . \tag{3.29}$$

It is assumed for the synthesis model of this work that the actuators realize the desired torques ideally.

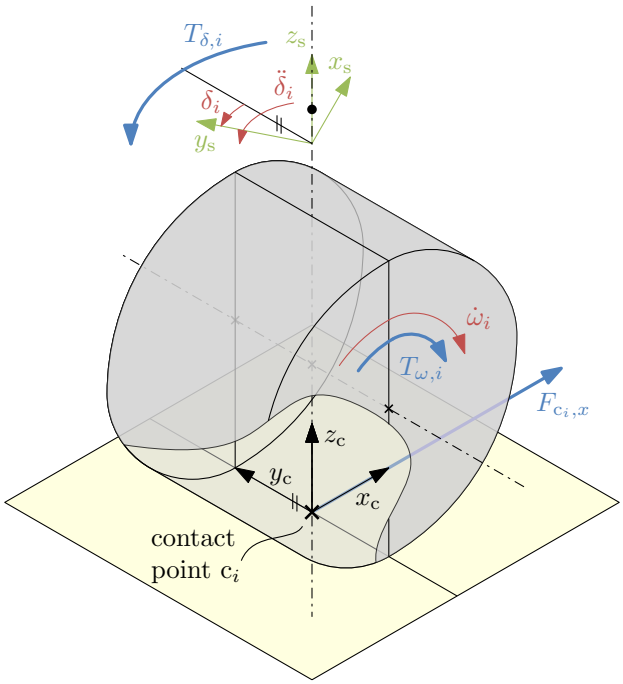


Figure 3.3: Quantities of the combined steering and drive dynamics.

3.5 State of the Art Terramechanics Modeling

The terramechanics, i.e. the interaction between wheels and soft soil, is the most challenging subsystem of a mobile robot to model. Soil flow underneath and around the contact patch, and even soil transport into or over the wheel, occurs and has a major impact on the contact forces. To model these disruptive effects physically, the computationally expensive Discrete Element Method (DEM) is necessary. A Finite Element Method (FEM) assumes a continuum and hence cannot represent soil rupture, however, it still allows soil states in all three spatial dimensions. For a further simplification, Krenn and Hirzinger [KH09] and Buse [Bus18] developed SCM, a height-map based approach with explicit computations only. The authors use an equally distributed 2D mesh of the ground, where soil height, and other states, are computed and stored per node. This allows for a limited soil flow representation and is referred to as a 2,5D approach. While the SCM is significantly more efficient than a DEM, it is still around 20 times slower than real-time for the simulation of a full rover on an office computer. Due to even longer computation times on a less powerful onboard computer, SCM is not suitable for onboard application.

For any time-sensitive terramechanics computation, further simplifications are therefore needed and semi-empirical models, that are based on the work of Bekker, Reece, Janosi, Hanamoto and Wong, are mostly used [Won08]. The different model types are summarized in Table 3.2 and more detailed information about the methods as well as their advantages and disadvantages can be found in [Lic+16].

In the following, the notation for the contact forces is presented in Section 3.5.1, before introducing the semi-empirical model as the most popular terramechanics modeling concept

Table 3.2: Overview of terramechanics model types.

	Discrete Element Method (DEM)	Soil Contact Model (SCM)	Semi-empirical models
Characterization	Particle-particle contact with physical contact force laws	Semi-empirical contact force based on 2.5D soil flow & deformation	Semi-empirical contact force through integration over contact patch
Soil states	✔	✔(2D)	✘
Soil deformation	✔	✔(empirical)	✘
Application example	High-fidelity single wheel simulation, wheel optimization	Full rover simulation with focus on terramechanics	Full rover simulation, optimization

in Section 3.5.2. Some remarks on existing shortcomings of the semi-empirical model are finally given, leading to a novel, purely empirical model in Section 3.6.

3.5.1 Contact Force Definition

In this work, the contact force of the ith contact point is divided into the tangential force $\boldsymbol{F}_{\mathrm{c}_i,\mathrm{T}}$ and the normal force $F_{\mathrm{c}_i,\mathrm{N}}$ as

$$\boldsymbol{F}_{\mathrm{c}_i} = \begin{bmatrix} \boldsymbol{F}_{\mathrm{c}_i,\mathrm{T}} \\ F_{\mathrm{c}_i,\mathrm{N}} \end{bmatrix} \quad \text{with} \quad \boldsymbol{F}_{\mathrm{c}_i,\mathrm{T}} = \begin{bmatrix} F_{\mathrm{c}_i,x} \\ F_{\mathrm{c}_i,y} \end{bmatrix}. \tag{3.30}$$

For each contact point, the tangential force can be controlled through the associated wheel drive and steering actuators and the force itself consists of longitudinal and lateral force $F_{\mathrm{c}_i,x}$ and $F_{\mathrm{c}_i,y}$, respectively. The normal force $F_{\mathrm{c}_i,\mathrm{N}}$, on the other side, is not controllable but results from gravity forces, the current rover configuration as well as the ground geometry and properties. For a separate handling due to these different controllability, not only the 3-dimensional contact force vectors $\boldsymbol{F}_{\mathrm{c}_i}$ are concatenated. Additionally, the tangential and normal contact forces are concatenated separately for the six contact points, yielding the three vectors

$$\boldsymbol{F}_{\mathrm{c,T}} = \begin{bmatrix} \boldsymbol{F}_{\mathrm{c}_1,\mathrm{T}} \\ \vdots \\ \boldsymbol{F}_{\mathrm{c}_6,\mathrm{T}} \end{bmatrix}, \quad \boldsymbol{F}_{\mathrm{c,N}} = \begin{bmatrix} F_{\mathrm{c}_1,\mathrm{N}} \\ \vdots \\ F_{\mathrm{c}_6,\mathrm{N}} \end{bmatrix} \quad \text{and} \quad \boldsymbol{F}_{\mathrm{c}} = \begin{bmatrix} \boldsymbol{F}_{\mathrm{c}_1} \\ \vdots \\ \boldsymbol{F}_{\mathrm{c}_6} \end{bmatrix}. \tag{3.31}$$

3.5.2 Semi-Empirical Terramechanics Models

The goal of this subsection is a brief description of semi-empirical terramechanics models following the concepts of Bekker, Wong, Janosi & Hanamoto. Subsequently, their suitability for onboard usage, in particular for controller synthesis, is discussed. It is not intended to provide an exhaustive treatment of the subject in this section, the interested

reader is again referred to the book of Wong [Won08] and to [Din+15b] for an overview of more recent work.

Bekker found an empirical relationship for the normal stress σ onto a plate that is pressed into soil to a depth d,

$$\sigma = \left(\frac{k_c}{b} + k_\phi \right) d^{k_n} . \tag{3.32}$$

Therein, b is the plate's smaller dimension and k_c, k_ϕ and k_n are pressure-sinkage parameters. This parameter set is sometimes called Bekker parameters and can be determined experimentally with a device called Bevameter, see e.g. [Apf+11]. For a rigid wheel, Bekker assumed that the normal pressure σ acts radial to the wheel rim and can be calculated for each contact point with (3.32), using the depth of the respective point as d. Once the wheel rotates, it shears the soil and thereby causes a tangential stress, which can grow up to the Mohr-Coulomb failure criterion

$$\tau_{\max} = \sigma \tan(\phi) + c . \tag{3.33}$$

The angle of internal resistance ϕ and the cohesion c are soil parameters and exceeding this maximum shear stress yields failure of the soil. Equation (3.33) states a linear relationship between the normal stress and the maximum shear stress. The offset c represents a bonding irrespective of the normal pressure for materials like saturated clay or special types of dry soil, e.g. chalk. For loose soil, Janosi and Hanamoto propose that the actual shear stress τ is an exponential function of the shear displacement j

$$\tau = \tau_{\max} \left(1 - e^{-j/k_j} \right) \tag{3.34}$$

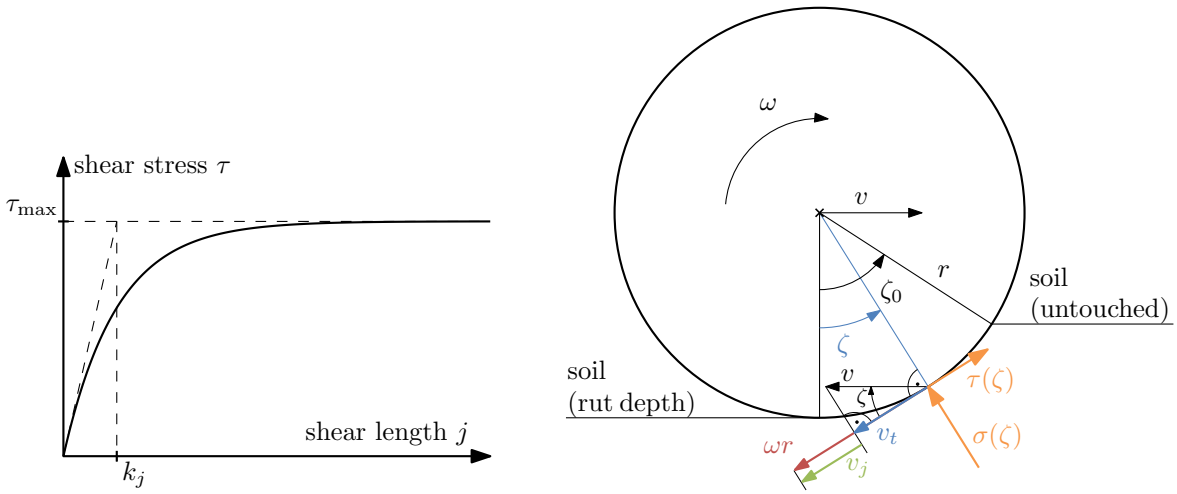
with the shear deformation parameter k_j. The shape of this function is depicted in Figure 3.4a, where the parameter k_j is the shear length value at the intersection between the tangent to the shear stress curve in the origin and the maximum shear stress. According to [Won08], k_j is approximately 2.5 cm for loose sand. By integrating both, the normal and shear stress, over the contact patch, forces in longitudinal and normal direction are obtained as

$$\begin{bmatrix} F_z \\ F_x \end{bmatrix} = rb \int_0^{\zeta_0} \begin{bmatrix} \cos(\zeta) & \sin(\zeta) \\ -\sin(\zeta) & \cos(\zeta) \end{bmatrix} \begin{bmatrix} \sigma(\zeta) \\ \tau(\zeta) \end{bmatrix} \mathrm{d}\zeta . \tag{3.35}$$

Herein the stresses are needed as functions of the angle ζ. For the normal stress, this is straightforward with Bekker's assumption that the sinkage in (3.32) is the depth of each individual point on the wheel rim, yielding

$$d(\zeta) = d_0 - r(1 - \cos(\zeta)) \tag{3.36}$$

with the maximum sinkage d_0 at $\zeta = 0$. For no slippage and a known normal force, this d_0 can be calculated. Therefore, the first line of (3.35) is solved for ζ_0 after setting $\tau = 0$ and the normal force F_N to a given wheel load. The found value for ζ_0 and $d(\zeta) = 0$ are then used in (3.36) and the equation is solved for d_0. When it comes to the shear stress as a function of the angle, $\tau(\zeta)$, the shear length j must be investigated in more detail. In



(a) Exemplary shape of the shear stress vs. shear length dependency according to Janosi and Hanamoto.

(b) Concept of sinkage, rut formation and definition of the angle ζ for the integration of stress over the contact patch.

Figure 3.4: Illustrations for semi-empirical soil modeling.

[Won08], a settled movement of the wheel, with a forward ground truth velocity v and the rotation rate ω, is assumed, see Figure 3.4b. The tangential velocity component at the rim as a function of ζ results as

$$v_t = v\cos(\zeta) = (1-i)r\omega\cos(\zeta) \tag{3.37}$$

with the wheel slip

$$i = 1 - \frac{v}{r\omega}\,. \tag{3.38}$$

The slip velocity v_j results as

$$v_j = r\omega - v_t = r\omega\left(1-(1-i)\cos(\zeta)\right) \tag{3.39}$$

and is integrated to finally obtain the slip length as a function of the angle ζ

$$\begin{aligned} j(\zeta) &= \int_0^t v_j \mathrm{d}t = \int_\zeta^{\zeta_0} r\omega\left(1-(1-i)\cos(\zeta)\right)\frac{\mathrm{d}\zeta}{\omega} \\ &= r\left[(\zeta_0-\zeta)-(1-i)(\sin(\zeta_0)-\sin(\zeta))\right]\,. \end{aligned} \tag{3.40}$$

To obtain the normal and shear stress as functions of the angle ζ, (3.40) is used in the shear stress equation (3.34) and (3.36) in the pressure-sinkage relation (3.32). The resulting stress equations can then be used in (3.35) to compute the normal and traction forces of the wheel.

According to [Din+15b], Wong and Reece found that the assumed exit angle $\zeta = 0$ is generally not correct and that the maximum stress does not occur at the deepest point of the wheel but further towards the entry angle ζ_0. Other researchers found more shortcomings of the above approach, such as inaccurate results for small vehicles [MS11], missing slip-sinkage [Din+10; Zho+14], additional forces from grousers and sidewall resistance [Zho+14] and a

difference between the angles of maximum normal and maximum shear stress [Gao+13]. All these authors introduce additional equations with tuning parameters to overcome the specific flaw they present.

For an onboard use, force laws need to be found that can be integrated analytically to prevent the necessity of spatial integration in the model. There are approaches to simplify the above equations to make explicit formulations possible [Din+15a; JSP13], however, the focus is always on a correct representation of the steady state. While the experimental findings that led to (3.34) are based on shearing soil to create shear force, the shear length j is commonly taken as a quantity of a point on the wheel rim as in (3.40). Using this shear length formulation for the force calculation in (3.35) yields a traction force which is purely dependent on the slip i, not on its history. It thus contains no additional information compared to a purely slip, i.e. velocity-based, formulation. Strictly speaking this shear length would need to be a state of the soil, requiring a spatial soil discretization and even then it is unclear how the shear length evolves for dynamic cases. This raises questions such as

- Is the stress, and thereby the traction force, purely dependent on the shear length or also on velocity and acceleration?
- How does the shear length develop for a wheel that stops, starts over or reverses?
- Is the shear length indeed a soil state, i.e. is it maintained and preceding wheels see immediate shear stress and/or different development of the shear stress compared to the first wheel, which drives in fresh soil?

Existing single wheel test campaigns do not answer these questions since they are always focused on the developed traction force for the steady state, i.e. after the acceleration and with constant wheel rate and slip. For the use in a dynamic controller, however, the importance of the synthesis models' dynamic behavior is obvious. These questions are a topic of ongoing research, including a new approach for a purely empirical, real-time capable model, see Section 3.6 and [Bar18]. Preparations for dynamic single wheel testing and a first test campaign were recently performed at the institute's Terramechanics Robotic Locomotion Lab (TROLL), see [BBL18] for a description of the lab.

3.6 TerRA: An Empirical, Real-time Capable Terramechanics Model

Due to the shortcomings of existing semi-empirical terramechanics models, that were listed in Section 3.5.2, a purely-empirical and real-time capable model is being developed at our institute. The approach, model details and first results of that work are summarized in this section and were published in [Bar18]. Following the purpose of this model, it is named Terramechanics for Real-time Application (TerRA). At this point, it consists of empirical

equations for plastic and dynamic sinkage and resulting normal and longitudinal forces. It is partly validated against the higher-fidelity SCM model.

3.6.1 Model Requirements

For making sure that the new model suits its purpose and for allowing a structured model development, modeling principles and desired effects were first established and are summarized in the following.

General Idea and Requirements. To allow for an onboard usage, the computational efficiency of TerRA is one major requirement. This prohibits any spatial discretization of the soil, as in SCM, or the contact patch, as used in most semi-empirical models, see Section 3.5. Derived from this restriction and the further considerations in Section 3.5, a decision for purely empirical modeling was made. The main idea is to cover observed effects from tests and high-fidelity models by simple mathematical relations, which contain a set of tuning parameters. In order to be of use, it must ultimately be possible to tune these parameters with said tests or complex simulation models for a certain wheel-soil combination. Without change of parameters, the empirical model must then be capable of reproducing contact forces for the full range of possible scenarios. In other words, the set of parameters needs to be dependent only on the wheel-soil combination, not on the driven scenario, e.g. speed, slope and acceleration.

Plastic sinkage. For contact with other objects, sandy soil mainly shows plastic deformation. When a wheel is dropped or pushed into the sand, this plastic deformation leads to a higher sinkage of the wheel, which is called plastic sinkage in this work. As long as the wheel stays in the same spot, this sinkage does not decrease anymore. It is therefore only by driving out of the area of plastic sinkage onto soil with less deformation, that the sinkage can be reduced. Multi-pass effects, i.e. trailing wheels drive through the ruts of the leading wheels with different soil geometry and states, cannot be covered without a soil discretization. However, the local effect of plastic sinkage onto one wheel has to be covered by the proposed model.

Slip-sinkage. Slipping wheels tend to sink into the soil further, especially grousered wheels in soft sand act much like a blade wheel for high slip. This effect is crucial since the resistance force, that opposes movement of the wheel, is mainly dependent on the sinkage depth. In contrast to a wheel being pushed into the sand, the sand is transported to the back of the wheel for slip-sinkage, therefore, there is no compaction and no increased normal force.

Slip and traction force. The traction force is highly dependent on the slip of a wheel. However, compared to the tire-road contact of road vehicles, a direct relationship without

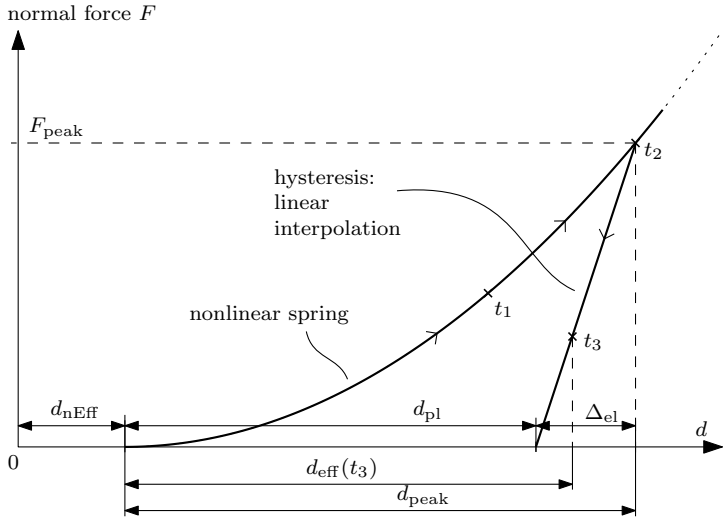


Figure 3.5: Different components of the geometric wheel sinkage and their influence on the normal force.

the consideration of the soil shear history is not accurate. Similar to the plastic sinkage, this soil shear history cannot be stored without a soil memory, i.e. spatial discretization. The local shear history shall still be considered by integration of the slip, which results in a dynamic development of the traction force and the slip. For non-changing conditions, the dynamic movement shall settle to a certain, non-zero slip value that has to be different for different wheel load, soil and external forces such as downhill force.

Resistance force. As mentioned above, the soil develops a resistance force, which opposes wheel movement, i.e. it can only counteract movement and not lead to acceleration. This force is dependent on the sinkage of the wheel and ranges from close to zero to a force that cannot be overcome anymore by traction. The latter case ultimately leads to embedding of the wheel if the sinkage gets too high.

Convergence. Although the above mentioned dynamic effects shall be covered by the model, it must converge to a stable steady state. A remaining drift or instability for non-changing external conditions would be non-physical and is therefore undesired.

3.6.2 Model Development

The model development of TerRA is described in detail in the following paragraphs. An explanation on different components of the geometrical wheel sinkage is given at the beginning, followed by the normal force, the slip-sinkage and the longitudinal force models.

Effective and Non-Effective Sinkage

For this model, the sinkage d of a wheel is defined as the geometric penetration of the assumed wheel contact point into the undeformed ground. When it comes to the penetration-based normal force calculation, which will be detailed in the next paragraph, the full sinkage d must be divided into two types. This is necessary to allow for the slip-sinkage phenomenon, which was described in Section 3.6.1. The two components are (see Figure 3.5):

- *Effective sinkage* (d_{eff}): This sinkage part consists of the elastic and plastic deformation of the ground underneath a non-slipping wheel. It is used for the normal force calculation.
- *Non-effective sinkage* (d_{nEff}): This sinkage results from a soil transport from under the wheel to its back by slippage, especially for grousered wheels. When assuming the soil parameters being homogeneous over the depth, this sinkage part does not influence the normal force, since the soil is physically moved, rather than compacted.

Both sinkages add up to the total geometric sinkage d as

$$d = d_{\mathrm{eff}} + d_{\mathrm{nEff}} \ . \tag{3.41}$$

Normal Force Model

The normal force model of TerRA is a standard elasto-plastic model with a nonlinear spring characteristic and was inspired by [LK16], which developed an elasto-plastic soil model for a self-impelling nail. In Figure 3.5, the force characteristics, as well as the different components of the sinkage, can be seen. For a descriptive understanding of the model, the behavior of a wheel that is dropped onto the ground is described by reference to the three times $t_1 - t_3$ in Figure 3.5:

- At time t_1, the wheel has already touched the ground and sinks further into the soil, while the normal force continues to build up according to its nonlinear characteristic.
- At time t_2, the rising normal force has slowed down the wheel enough, such that it does not sink any deeper; it has reached its peak sinkage d_{peak}. For a normal force model without hysteresis, the normal force would now decrease on the same nonlinear branch. However, due to the implemented hysteresis, the normal force drops much faster on the linear curve that is shown in Figure 3.5.
- At time t_3, the normal force and the wheel load and weight have reached an equilibrium and the wheel stays at its current sinkage. For changing wheel loads, the normal force now moves on the linear curve, which corresponds to a linear spring and is called elastic layer Δ_{el}. The plastic sinkage d_{pl} is indicated in Figure 3.5 as well and stays

constant as long as the wheel remains in the elastic layer. If the loads rise high enough to cause the wheel reaching its peak again, the normal force continues to increase on the nonlinear curve until the wheel has reached a new peak sinkage.

After this explanation, the model equations to represent the desired behavior are derived in the following. The derivative of (3.41) with respect to time yields the total sinkage rate

$$\dot{d} = \dot{d}_{\text{eff}} + \dot{d}_{\text{nEff}} \ . \tag{3.42}$$

The non-effective sinkage d_{nEff} does not influence the normal force and is therefore not needed within this paragraph. As described above, the peak sinkage stays constant if the effective sinkage is below the peak, which can be written as

$$\dot{d}_{\text{peak}} = \begin{cases} 0 & \text{for } d_{\text{eff}} < d_{\text{peak}} \ , \\ \dot{d}_{\text{eff}} & \text{else} \ . \end{cases} \tag{3.43}$$

The associated peak normal force results with its nonlinear characteristic

$$F_{\text{peak}} = k_{\text{s}} d_{\text{peak}}^{(k_{\text{exp}})} \ . \tag{3.44}$$

with the spring constant k_{s} and exponent k_{exp}. Within the elastic layer, the normal force is calculated as the linear spring force with stiffness k_{el} as

$$F_{\text{spring}} = k_{\text{el}} \left(z_{\text{eff}} - z_{\text{pl}} \right) \ . \tag{3.45}$$

The plastic sinkage, as can easily be seen in Figure 3.5, results as

$$d_{\text{pl}} = d_{\text{peak}} - \Delta_{\text{el}} \tag{3.46}$$

with the elastic layer of thickness Δ_{el} resulting from its constant stiffness k_{el},

$$\Delta_{\text{el}} = \frac{F_{\text{peak}}}{k_{\text{el}}} \ . \tag{3.47}$$

To replicate the energy dissipation of sandy soil, a linear damping force

$$F_{\text{damping,N}} = k_{\text{d}} 2 \sqrt{k_{\text{s}} m} \cdot \max \left(\dot{d}_{\text{eff}}, 0 \right) \tag{3.48}$$

is added to the normal spring force with the damping ratio k_{d} and the estimated load mass on the wheel m. The 'max' term in (3.48) ensures, that only a rising sinkage is damped.

The total normal force is the sum of spring and damping force,

$$F_{\text{N}} = F_{\text{spring}} + F_{\text{damping}} \ . \tag{3.49}$$

Dynamic sinkage

Most slip-sinkage approaches for semi-empirical terramechanic models in the literature vary the sinkage exponent k_n in (3.32), see e.g. [Din+10; Zho+14]. Through the integration of the normal stress to the contact forces, this variation indirectly influences the resulting sinkage of the wheel. In the present model, however, the slip-sinkage is explicitly modeled as a non-effective sinkage. It is not taken into account for the normal force computation, as was shown in the previous paragraphs.

The sinkage in this model increases with the longitudinal slip velocity of the wheel

$$v_{\mathrm{sl}} = \omega r - v_{\mathrm{c}|0,x} \ . \tag{3.50}$$

It is the difference between the wheel rim velocity and the actual longitudinal velocity of the wheel. For the slip-sinkage rate $\dot{d}_{\mathrm{sls}}$, a nonlinearly sinkage-dependent coefficient k_{sls} is used such that

$$\dot{d}_{\mathrm{sls}} = k_{\mathrm{sls}} |v_{\mathrm{sl}}| \quad \text{with} \quad k_{\mathrm{sls}} = k_{\mathrm{sls,max}} \left(\frac{\max\left(k_{\mathrm{sls},0} - d, 0\right)}{k_{\mathrm{sls},0}} \right)^{k_{\mathrm{sls,exp}}} . \tag{3.51}$$

The parameter $k_{\mathrm{sls,max}}$ is the maximum coefficient between slip velocity and slip-sinkage rate. $k_{\mathrm{sls},0}$ represents the total wheel sinkage at which the slip-sinkage goes to zero irrespective of the slip velocity. The exponent $k_{\mathrm{sls,exp}}$ finally allows for a nonlinear function between zero and $k_{\mathrm{sls,max}}$.

So far, the plastic sinkage d_{pl} from (3.46), as well as the slip-sinkage d_{sls} with its rate (3.51), can only increase. This is sensible as long as the wheel remains in one spot, since neither the compressed soil nor the soil that has been transported to the back of the wheel recovers or flows back. For a tangential movement of the wheel, however, it drives onto new soil, which is less compressed and has not been shoveled away. In terms of the dynamic sinkage, this equals an outward sinkage rate $\dot{d}_{\mathrm{out}}$. This rate is subtracted in parts from the peak sinkage rate $\dot{d}_{\mathrm{peak}}$, and thereby lowers the plastic sinkage, and the slip sinkage rate $\dot{d}_{\mathrm{sls}}$. For the computation of the current total outward sinkage rate, it is assumed that the wheel can reach its minimum sinkage d^* by moving half the contact patch length forward, see Figure 3.6. Therein, d^* is the sinkage that it needs to support the wheel load statically. This minimum sinkage is calculated by equating the nonlinear spring force F_{peak} and the load of the wheel mg and inversion of (3.44), resulting in

$$d^* = \left(\frac{mg}{k_{\mathrm{s}}} \right)^{\left(\frac{1}{k_{\mathrm{exp}}}\right)} . \tag{3.52}$$

Assume a wheel that experienced large loads and/or high slip-sinkage and therefore has a large total sinkage. By traction or external forces, it starts to move forward. The above assumption means that it can get rid of its accumulated additional sinkage as soon as the contact point reached untouched soil. The geometric relation is depicted in Figure 3.6 and

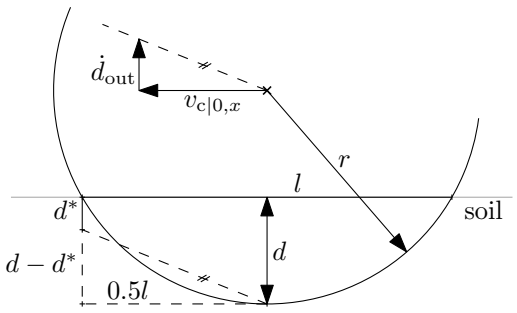


Figure 3.6: Visualization of the outward dynamic sinkage calculation.

through geometric similarity, the outward sinkage rate is described by the equation

$$\dot{d}_{\mathrm{out}} = \min\left(\frac{(d-d^*)}{0.5l}, \tan\left(\phi_{\mathrm{G}}\right)\right)\left|v_{\mathrm{c}|0,x}\right| . \tag{3.53}$$

Due to the granular property of the soil, its piling angle is limited. For non-cohesive, dry sand this angle corresponds approximately to the gravity-corrected internal friction angle

$$\phi_{\mathrm{G}} = \phi - \alpha, \tag{3.54}$$

with the inclination of the soil plane in longitudinal wheel direction α. In (3.53), the virtual ramp, that the wheel is able to drive out of the soil on, is limited to this angle. The contact patch length l is obtained through

$$l = 2\sqrt{2rd - d^2} \tag{3.55}$$

with the radius r and the total sinkage d. The total outward dynamic sinkage rate $\dot{d}_{\mathrm{out}}$ from (3.53) is divided into the plastic sinkage part and the slip-sinkage part as

$$\dot{d}_{\mathrm{out}} = \dot{d}_{\mathrm{out,peak}} + \dot{d}_{\mathrm{out,nEff}} . \tag{3.56}$$

The term $\dot{d}_{\mathrm{out,peak}}$ reduces the plastic wheel sinkage and is assumed linear within the elastic layer:

$$\dot{d}_{\mathrm{out,peak}} = \min\left(\max\left(\frac{d_{\mathrm{peak}} - d_{\mathrm{eff}}}{\Delta_{\mathrm{el}}}, 0\right), 1\right)\dot{d}_{\mathrm{out}} . \tag{3.57}$$

Therein, the reduction rate of the plastic sinkage goes to zero as the actual sinkage approaches the peak sinkage. If this outward peak sinkage rate is smaller than the total outward sinkage rate $\dot{d}_{\mathrm{out,peak}} < \dot{d}_{\mathrm{out}}$, the remaining rate from (3.56) reduces the non-effective sinkage according to

$$\dot{d}_{\mathrm{nEff}} = \dot{d}_{\mathrm{sls}} - \dot{d}_{\mathrm{out,nEff}} . \tag{3.58}$$

Ultimately, the peak fraction of the outward sinkage rate needs to be added to the peak sinkage rate equation from (3.43) if the wheel is in the elastic layer, leading to the final peak sinkage rate

$$\dot{d}_{\mathrm{peak}} = \begin{cases} \dot{d}_{\mathrm{out,peak}} & \text{for} \quad d_{\mathrm{eff}} < d_{\mathrm{peak}} \\ \dot{d}_{\mathrm{eff}} & \text{else.} \end{cases} \tag{3.59}$$

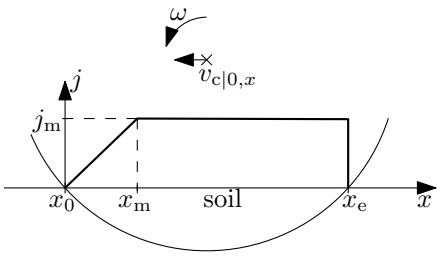


Figure 3.7: Development of the slip length j over the wheel contact length x.

Longitudinal Force Calculation

The longitudinal force model of TerRA is based on JANOSI and HANAMOTO's empirical shear stress equation (3.34). Towards the end of Section 3.5.2, a problem with the computation of the shear length state j was, however, stated. Since the relation represents soil shear behavior, the shear length would need to be a state of discretized soil patches. A soil discretization is not feasible within the fast TerRA model, therefore a novel approach for covering the local shear behavior, see *Slip and traction force* in Section 3.6.1, was developed.

First, a simplified development of shear stress over the length of the contact patch is assumed, see Figure 3.7, consisting of a linear and a constant part. Moving to the left in the figure, the wheel hits new soil at the coordinate x_0, therefore the soil has not been sheared at that point, i.e.

$$j(x_0) = 0 \; . \tag{3.60}$$

From this soil entry point, the shear length development is simplified with a constant gradient until it reaches the point (x_m, j_m). At that point, the maximum shear length of the current contact situation is reached and $j(x)$ stays constant within $[x_m, x_e]$. The maximum shear length value j_m is obtained as the time integral over the slip velocity from (3.84)

$$j_m = \int v_{sl} \mathrm{d}t = \int \omega(t) r - v_{c|0,x}(t) \mathrm{d}t \tag{3.61}$$

To enable a dynamic change of the contact situation, even when the maximum slip length is fully developed, the position x_m is dynamic and computed through a first order filter as

$$\dot{x}_m = \left(\frac{v_{c|0,x}}{v_{sl}} j_m - x_m \right) / T \tag{3.62}$$

with the time constant T.

For a better understanding of the resulting model, Figure 3.8 shows the slip length over the contact patch for zero (left) versus relatively high (right) translational velocity of the wheel. Due to the resulting traction force corresponding to the integral of the slip length over the contact patch, it corresponds to the area below the j-curve. It can be seen in Figure 3.8 that this area shrinks considerably for high translational velocities, especially since the point x_m is not limited to the contact patch but can grow higher than x_e. That case of $x_m > x_e$ is indicated on the right side of Figure 3.8: The integral yields only the triangular area above

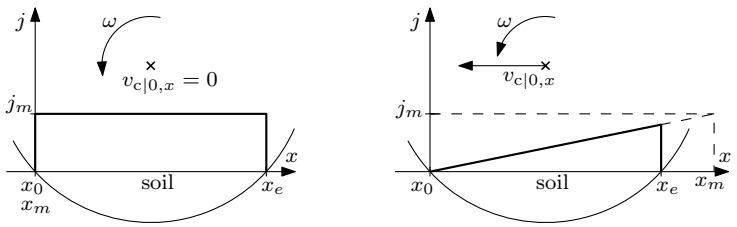


Figure 3.8: Development of the slip length j over the contact patch for zero (left) versus high (right) translational wheel velocity.

the interval $[x_0; x_\mathrm{e}]$, since values of $x > x_\mathrm{e}$ are outside of the contact patch and thus not considered. Therefore, the area, and thereby the traction force, approaches zero as x_m goes to infinity.

With these relations for the slip or shear length j, the Janosi-Hanamoto shear stress from (3.34), which was plotted in Figure 3.4a, can be computed. For the traction force, this stress is integrated over the contact patch with the assumptions of a rectangular contact patch and a constant stress over the wheel width. The resulting integral can be solved analytically as

$$\begin{aligned} F_\mathrm{tr} &= \int_{x_0}^{x_\mathrm{e}} \int_{-w/2}^{w/2} \tau_\mathrm{max} \left(1 - e^{-(j(t)/k_j)}\right) \,\mathrm{d}y \,\mathrm{d}x \\ &= w\tau_\mathrm{max} \begin{cases} \left[x_\mathrm{e} + x_\mathrm{m} \dfrac{k_j}{j_\mathrm{m}} \left(e^{-(j_\mathrm{m} x_\mathrm{e})/(x_\mathrm{m} k_j)} - 1\right)\right] & \text{for } x_\mathrm{m} > x_\mathrm{e} \\ \left[x_\mathrm{e}\left(1 - e^{-(j_\mathrm{m}/k_j)}\right) + x_\mathrm{m}\left(\dfrac{k_j}{j_\mathrm{m}}\left(e^{-(j_\mathrm{m}/k_j)} - 1\right) - e^{-(j_\mathrm{m}/k_j)}\right)\right] & \text{else.} \end{cases} \end{aligned} \tag{3.63}$$

Additionally to this traction force, the wheel sinkage causes a maximum soil resistance force F_res, which opposes any translational movement of the wheel. One approximation of that force is with the equation of passive earth pressure under the assumption that the wheel with sinkage d acts like a smooth, vertical surface that penetrates the soil to the depth d. When being pushed through the soil, the resistance force can be calculated as [TPM96]

$$F_\mathrm{res} = 0.5\rho g d^2 \left(\tan^2\left(\frac{\pi}{4} + \frac{\phi}{2}\right)\right) + 2cd\tan\left(\frac{\pi}{4} + \frac{\phi}{2}\right) \tag{3.64}$$

with the soil density ρ, the gravity g, the internal soil friction angle ϕ and the cohesion c.

The final longitudinal force $F_{c_i,x}$ is calculated as the sum of the regularized traction force F_tr from (3.63) and the resistance force. Due to F_res from (3.64) being the maximum resistance force that can act in the opposite direction of the longitudinal wheel velocity, it needs to be regularized as well. While the traction force is simply regularized with a hyperbolic tangent, a PI-controller is used to get rid of a remaining sliding velocity and still ensure stability with a fast, fixed step solver. With anti-windup, the controlled resistance force reads

$$\tilde{F}_\mathrm{res} = k_\mathrm{P}\left(v_{c|0,x} F_\mathrm{res} + k_\mathrm{I} \int \left(v_{c|0,x} F_\mathrm{res} - \frac{1}{k_\mathrm{P} k_\mathrm{aw}} \Delta_\mathrm{lim}\right) \,\mathrm{d}t\right) \tag{3.65}$$

with the proportional gain k_P, the integral gain k_I, the anti-windup gain k_aw and the difference Δ_lim between $\tilde{F}_\mathrm{res}$ and the limit F_res in case of exceeding or falling below the limits. The final tangential force results as

$$F_{\mathrm{c}_i,x} = F_\mathrm{tr} \tanh\left(-\frac{v_\mathrm{sl}}{k_v}\right) + \begin{cases} F_\mathrm{res} & \text{for } \tilde{F}_\mathrm{res} > F_\mathrm{res} \\ -F_\mathrm{res} & \text{for } \tilde{F}_\mathrm{res} < -F_\mathrm{res} \\ \tilde{F}_\mathrm{res} & \text{else} \end{cases} \tag{3.66}$$

with k_v being a parameter to stretch the hyperbolic tangent.

A short note on the effect of grousers concludes the TerRA model development. Some researchers, e.g. [Zho+14], extended the semi-empirical modeling from Section 3.5.2 by additional traction force models that are based on assuming the grousers as plates that are pushed through the soil. Equations for this additional grouser force are based on earth pressure, similar to the resistance force (3.64) in this work. However, the grousers mainly have two effects: First, the average traction force of a grousered wheel is higher and, second, the grousers entering and exiting the soil while the wheel is turned leads to an oscillating traction force. According to [JSP13], the average increase of traction force can be approximated by introducing an equivalent wheel radius, which is higher for larger grousers. For the dynamic variations of the traction force due to the grousers, [IBW11] proposed adding oscillations to the average force. Coming back to the TerRA model, the average effect of grousers is implicitly considered by tuning the model parameters to more complex models or experiments that include the grouser effects. An oscillation to model the dynamic grouser effect, similar to [IBW11], could easily be added to TerRA in the future if needed.

3.6.3 Qualitative Verification of the Model Requirements

To conclude the section on the TerRA model, a simple simulation scenario with its results shall demonstrate to what extend the requirements from Section 3.6.1 are fulfilled. Furthermore, the following descriptions will help for a better understanding of the model behavior. For the wheel and model parameters that were used for this scenario, as well as further examples, a comparison to a high-fidelity terramechanics model and guidelines for tuning the TerRA parameters, the reader is referred to the original publication of this model [Bar18].

At $t = 0\,\mathrm{s}$, the wheel is dropped onto flat ground and starts turning at $t = 2\,\mathrm{s}$ with a constant angular velocity of $0.5\,\frac{\mathrm{rad}}{\mathrm{s}}$. A constant force of $10\,\mathrm{N}$ acts opposed to the driving direction of the wheel. The ground geometry has distinct kinks from 0° to 8° and from 8° to 20°, which are reached by the wheel at $t \approx 9\,\mathrm{s}$ and $t \approx 22\,\mathrm{s}$, respectively. The translational movements of the wheel in vertical and longitudinal direction are not restricted.

A detailed description of the slip and sinkage results in Figure 3.9 follows, where keywords of the requirements from Section 3.6.1 are *emphasized*. The drop of the wheel at $t = 0\,\mathrm{s}$ results in a *plastic sinkage*, which is part of the effective sinkage (red line in the right plot of Figure 3.9). This sinkage stays constant until $\approx 2.5\,\mathrm{s}$, where the wheel slowly starts to move

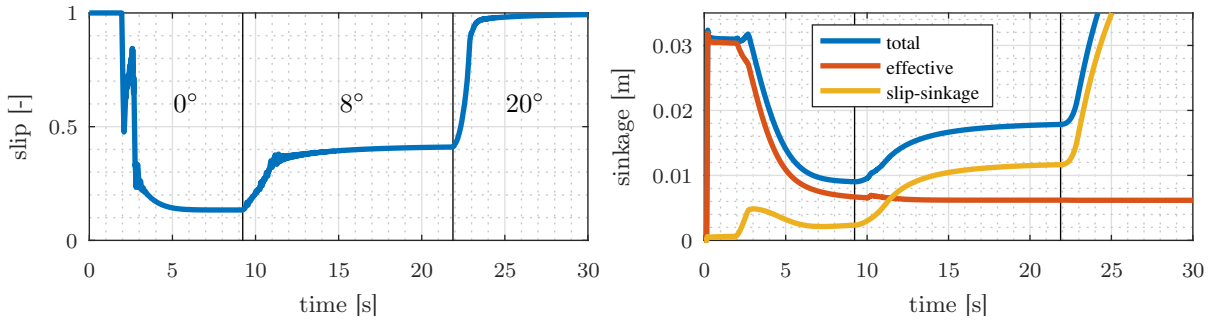


Figure 3.9: Wheel slip (left) and sinkages (right) for a single wheel driving on different slopes.

translationally, i.e. out of its plastic sinkage dip. As said above, the wheel starts rotating at $t = 2\,\mathrm{s}$ already. However, it can be seen in terms of a high slip in the left plot of Figure 3.9 that the wheel does not start to move in its longitudinal direction right away. This is due to the *slip and traction force* relationship, resulting in a slow built-up of the traction force that is needed to overcome the *resistance force* of the soil. The latter force is relatively high due to the high plastic sinkage from the drop. *Slip-sinkage* results from the initially high slip rate, rising between $t = 2\,\mathrm{s} - 3\,\mathrm{s}$, which is plotted as the yellow line in the right plot of Figure 3.9. From $\approx 3\,\mathrm{s}$ on, the wheel accelerates in its longitudinal direction, since the *traction force* overcomes the *resistance force*. The *dynamic sinkage* model yields an outward sinkage rate for a moving wheel, which can easily be observed in the fast drop of the *plastic sinkage* and the *slip-sinkage* (red and yellow curve in Figure 3.9).

This decrease continues until the wheel hits the first kink at $t \approx 9\,\mathrm{s}$. To overcome the additional downhill force, a higher traction force is needed, which can only be achieved with a higher slip according to the *slip and traction force* relation. The higher slip rate can be seen in the left plot of Figure 3.9 and leads to increased *slip-sinkage*. Although the dynamic processes take a few seconds, the slip and thereby the sinkage *converge* eventually. At $t \approx 22\,\mathrm{s}$, the wheel hits the second kink and needs even more traction force. As for the first kink, this is generated by more slip, however, in contrast to the 8° slope, the wheel cannot generate enough traction force to climb the 20° slope. To be more precise, the increased downhill force leads to more slip and thereby slip-sinkage. The higher sinkage then results in an increased resistance force, which ultimately leads to a wheel slip of 100 % and stops the wheel.

When it comes to the run time, this simulation ran roughly 100-times *faster than real-time* with a step size of 1 ms on a standard office computer.

In [Bar18], a systematic approach for parameter tuning and a first quantitative verification against the SCM model were shown. While the sinkage model allowed a good match, some discrepancies to SCM's dynamic force behavior were revealed. However, existing validation campaigns, including the ones for SCM, mostly focus on the steady-state force and sinkage and not their dynamic development. Therefore, comprising test campaigns need to be done and details about this problem and the testing capabilities of our institute were summarized

at the end of Section 3.5.2. More technical details about possible further work for TerRA can also be found in [Bar18].

3.6.4 Conclusion for the Controller Synthesis

In this Section 3.6, a novel fast terramechanics model was presented, that has two main advantages compared to the semi-empirical models from Section 3.5.2:

1. Some robot wheel specific and dynamic effects are built-in from the beginning rather than adding them to a model that was initially developed for static assessment of heavy offroad vehicles. This ultimately leads to less parameters and simpler equations.
2. No spatial discretization and only explicit equations enables a short run time of the model.

When it comes to the controller synthesis of this work, the TerRA model equations are not directly usable. A feedback linearization is used in this work, which implies inversion of a matrix that consists of Lie derivatives of the model equations, see Section 2.4. Therefore, the used model must be continuously differentiable. Looking at the model equations in Section 3.6.2, it can easily be seen that it contains several non-differentiable elements, such as a hysteresis, a limiter and a case differentiation. Further investigations need to be done whether simplifications can make TerRA suitable for the synthesis or if it is more sensible to use the model in forward mode for a parameter estimation, see Section 6.2.

3.7 Simplified Contact Model for Controller Synthesis

As described in Section 3.6.4, the terramechanics models that have been shown in Sections 3.5.2 and 3.6 are not suitable for a controller synthesis in this work. Therefore, simplified contact models for the normal and tangential forces are developed in this section. Two different models are presented for the normal forces: First, a spring-damper force model in Section 3.7.1, which contains non-measurable quantities but is needed in a stability analysis of the controller in Section 4.2.2. Second, a quasi-static estimation of the normal forces, which only requires measurable quantities, is derived in Section 3.7.2 for the usage in the controller. The tangential force model is based on the slip velocity and is presented in Section 3.7.3. For the general notation of the contact forces that is used in this work, the reader is referred to Section 3.5.1.

3.7.1 Sinkage-Based Normal Forces

Bekker's formulation for the normal stress (3.36) and other variations of it [Won08; Din+15b] are based on a sinkage depth d. By multiplying the stress with the current con-

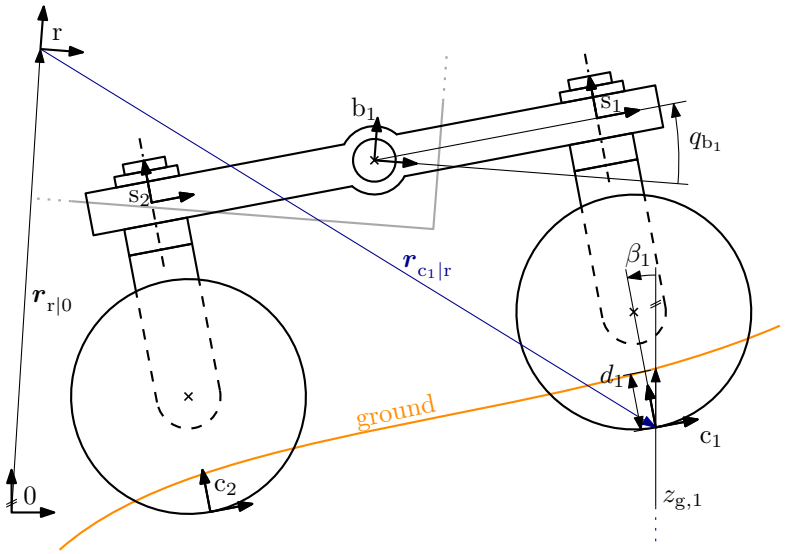


Figure 3.10: Coordinate systems and the geometric relations of a wheel with sinkage.

tact patch area, which is a function of the sinkage as well, a nonlinear spring normal force results. Although it is not considered in BEKKER's formula, soft soil introduces considerable damping which is captured by more complex models, e.g. SCM. For this section, the normal force is simplified to a linear spring-damper element with a dependency on the sinkage d_i of the ith wheel and its derivative (see Figure 3.10),

$$d_i = \cos(\beta)\left[z_{\mathrm{g},i} - \begin{bmatrix}0 & 0 & 1\end{bmatrix}\left(\boldsymbol{r}_{\mathrm{r}|0} + {}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}(\boldsymbol{\theta}_{\mathrm{r}})\boldsymbol{r}_{\mathrm{c}_i|\mathrm{r}}(\boldsymbol{q}_{\mathrm{b}})\right)\right] , \tag{3.67}$$

$$\dot{d}_i = \cos(\beta)\left[\dot{z}_{\mathrm{g},i} - \begin{bmatrix}0 & 0 & 1\end{bmatrix}\left({}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}(\boldsymbol{\theta}_{\mathrm{r}})\left(\boldsymbol{v}_{\mathrm{r}} + \underline{\boldsymbol{S}}(\boldsymbol{\omega}_{\mathrm{r}})\boldsymbol{r}_{\mathrm{c}_i|\mathrm{r}}(\boldsymbol{q}_{\mathrm{b}})\right)\right)\right] . \tag{3.68}$$

The absolute soil height at the ith wheel in the global '0'-frame is $z_{\mathrm{g},i}$ and $\dot{z}_{\mathrm{g},i}$ its derivative. $\boldsymbol{r}_{\mathrm{r}|0}$ is the rover position in the inertial frame and $\boldsymbol{r}_{\mathrm{c}_i|\mathrm{r}}(\boldsymbol{q}_{\mathrm{b}})$ the vector from the rover frame 'r' to the ith contact frame 'c_i'. β_i is the angle between the gravity vector and the ground normal, i.e. between the z-axes of the inertial '0'-frame and the ith contact frame 'c_i', see Figure 3.10. The translational and rotational velocities ${}^{\mathrm{r}}\boldsymbol{v}_{\mathrm{r}|0}$ and ${}^{\mathrm{r}}\boldsymbol{\omega}_{\mathrm{r}|0}$ as well as the RPY angles $\boldsymbol{\theta}_{\mathrm{r}}$ are the rigid body states from Section 3.3.

With the spring and damper constants k_{s} and k_{d}, respectively, the ith wheel normal force results as

$$F_{\mathrm{N}_i,\mathrm{springDamper}}(\boldsymbol{v}_{\mathrm{r}}, \boldsymbol{\omega}_{\mathrm{r}}, \boldsymbol{q}_{\mathrm{b}}, r_{\mathrm{r}|0,z}, z_{\mathrm{g},i}, \dot{z}_{\mathrm{g},i}) = k_{\mathrm{s}}d_i + k_{\mathrm{d}}\dot{d}_i . \tag{3.69}$$

Besides the measurable velocities of the rover, the spring-damper normal force (3.69) is dependent on the unknown soil height and the unmeasurable rover z-position. Due to these two unmeasurable disturbances having a major impact on the forces, this model cannot be used sensibly in the controller synthesis. It will still be needed later for stability analysis, as it relates the actual rover position with the normal forces, which the upcoming normal force estimation does not.

3.7.2 Quasi-Static Estimation of Normal Forces

It has been discussed in the last section, that a spring-damper normal force model, just like any penetration-based contact model, needs knowledge of two unmeasurable quantities. The normal forces are hence considered as disturbances and estimated directly for the controller in this work. Due to the slow motion of the ExoMars, the normal force estimation is computed with a quasi-static force and torque balance. Therefore, the ExoMars is assumed to stand still in its current configuration with its current bogie angles $\boldsymbol{q}_\mathrm{b}$ and its current orientation, which is represented by the RPY angles $\boldsymbol{\theta}_\mathrm{r}$. Furthermore, only forces in the direction of the gravity are considered, which is reasonable as long as tangential contact forces are relatively low compared to the normal forces. To decrease the errors for situations in which the tangential forces grow larger, they would need to be taken into account for the estimation. Further investigation could be directed towards either including the estimation into the model itself or using the tangential forces of the last timestep to prevent algebraic loops in the implementation, see also Section 6.2. Note however that the inclination and bogie angles are taken into account through the angle β in Figure 3.10 as will be seen towards the end of this subsection.

Due to the quasi-static assumption, the scalar force equilibrium between rover gravitational force and the bogie forces in the direction of the gravity $F_{\mathrm{g,b}_j}$, must be fulfilled,

$$\sum_{j=1}^{3} F_{\mathrm{g,b}_j} - g m_\mathrm{r} = 0 \,, \tag{3.70}$$

with the rover mass m_r and the gravity constant g. At each bogie j, the force $F_{\mathrm{g,b}_j}$ must be compensated by forces at the contact points of the two wheels that are mounted to the respective bogie. Therefore, one additional force equilibrium per bogie reads

$$F_{\mathrm{g,b}_j} - F_{\mathrm{g,c}_{j_1}} - F_{\mathrm{g,c}_{j_2}} = 0 \,, \quad j = 1 \dots 3 \,, \tag{3.71}$$

where the wheels j_1 and j_2 are attached to the jth bogie.

To derive the torque balances, the direction of the gravitational bogie forces must be considered with the z-axis unit vector ${}^0\boldsymbol{e}_{z_0} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$. The gravitational force can be expressed in the 'r'-frame by a rotation,

$${}^\mathrm{r}\boldsymbol{F}_\mathrm{g} = -g m_\mathrm{r} \, {}^\mathrm{r}\underline{\boldsymbol{R}}_0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = -g m_\mathrm{r} \, {}^\mathrm{r}\boldsymbol{e}_{z_0} \,. \tag{3.72}$$

For zero rotational acceleration of the rover, the torque equilibria around the x- and y-axes of the '0'-frame must yield zero as well,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} {}^0\underline{\boldsymbol{R}}_\mathrm{r} \sum_{j=1}^{3} \left(\boldsymbol{r}_{\mathrm{b}_j|\mathrm{r}} \times {}^\mathrm{r}\boldsymbol{e}_{z_0} \right) F_{\mathrm{g,b}_j} = \boldsymbol{0} \,. \tag{3.73}$$

The bogies are assumed frictionless, therefore the torque around each bogie's joint axis must be zero for zero angular acceleration, which leads to three more equations,

$$\begin{bmatrix}0 & 0 & 1\end{bmatrix} {}^{\mathrm{b}_j}\underline{\boldsymbol{R}}_{\mathrm{r}} \left({}^{\mathrm{r}}\boldsymbol{r}_{\mathrm{c}_{j_1}|\mathrm{b}_j} \times {}^{\mathrm{r}}\boldsymbol{e}_{z_0} F_{\mathrm{g},\mathrm{c}_{j_1}} + {}^{\mathrm{r}}\boldsymbol{r}_{\mathrm{c}_{j_2}|\mathrm{b}_j} \times {}^{\mathrm{r}}\boldsymbol{e}_{z_0} F_{\mathrm{g},\mathrm{c}_{j_2}} \right) = 0 \,, \quad j = 1 \ldots 3 \,. \tag{3.74}$$

Equations (3.70), (3.71), (3.73) and (3.74) add up to nine linearly independent scalar equations, to be solved for the 6 unknown contact and 3 unknown bogie forces.

Due to the scalars $F_{\mathrm{g},\mathrm{c}_i}$ all acting in the direction of the gravity, the vector of the gravitational force at the ith contact point reads

$$\boldsymbol{F}_{\mathrm{g},i} = F_{\mathrm{g},\mathrm{c}_i} \boldsymbol{e}_{z_0} \,. \tag{3.75}$$

The desired contact normal force, however, is the contact force component in the vertical z-axis of the 'c_i' frame, which is the assumed ground contact normal vector. Note that due to the zero contact angle assumption, the z-axes of each 's_i'-frame and its corresponding 'c_i'-frame are identical. Using the dot product of the two unit vectors, $\langle \boldsymbol{e}_{z_0}, \boldsymbol{e}_{z_{\mathrm{c}_i}} \rangle$, yields the final quasi-static normal force estimation

$$F_{\mathrm{N}_i,\mathrm{quasiStatic}}(\boldsymbol{\theta}_{\mathrm{r}}, \boldsymbol{q}_{\mathrm{b}}) = \cos(\beta_i) F_{\mathrm{g},\mathrm{c}_i} = \langle \boldsymbol{e}_{z_0}, \boldsymbol{e}_{z_{\mathrm{c}_i}}(\boldsymbol{\theta}_{\mathrm{r}}, \boldsymbol{q}_{\mathrm{b}}) \rangle F_{\mathrm{g},\mathrm{c}_i}(\boldsymbol{\theta}_{\mathrm{r}}, \boldsymbol{q}_{\mathrm{b}}) \,. \tag{3.76}$$

It can be seen, that the quasi-static force estimation only requires measurable quantities, i.e. the RPY angles $\boldsymbol{\theta}_{\mathrm{r}}$ and bogie angles $\boldsymbol{q}_{\mathrm{b}}$ as opposed to the spring-damper model (3.69).

3.7.3 Tangential Force

As described in Section 3.6.4, the terramechanics models from Sections 3.5.2 and 3.6 are not usable for the controller synthesis. Therefore, a one-point tangential force model, that is based on velocities, is developed in this section. Assuming a relation like Mohr-Coulomb's failure criterion (3.33) for the forces yields

$$F_{\mathrm{friction}} = F_{\mathrm{N}} \tan(\phi) \,. \tag{3.77}$$

The cohesion was therein assumed zero, since it can be neglected for dry sand [Won08]. To determine the only remaining parameter ϕ, it is important to understand what it actually describes. Considering the friction between solid objects, this angle is the tilt at which maximum static friction is overcome and an object starts sliding. In the theory of granular material, the angle ϕ is the inner shear angle. For dry, non-cohesive sand, the angle of repose, i.e. the steepest possible angle of piles of the granular material, is oftentimes used as an estimate for ϕ.

For the simplified one-point model of this work, the macroscopic effects, that this angle needs to reflect, are examined. In case of a contact of flat metal with sand, the friction angle is $\phi_{\text{M-S}} \approx 15°$, which is relatively low compared to the sand's internal friction. To increase the

traction, planetary exploration wheels are usually equipped with *grousers* that act similar to lugs in off-road tires by achieving partial form lock between wheel and soil. Depending on the shape of the grousers and the type of soil, this can lead to shear forces of up to the internal friction angle of sand, which is about $\phi_{\text{S-S}} \approx 25°\text{-}45°$ for sandy soil [Won08; AA18]. In fact, the grousers aim at achieving the maximum friction force, i.e. the maximum shear stress that the soil can bear before failure according to the MOHR-COULOMB criterion. At the end of Section 3.6.2, publications about the modeling of grouser effects were summarized. Some authors developed semi-empirical models, others derived an average effect and added oscillations. However, the average effect of the grousers is implicitly included in (3.77) when choosing a sand-typical value for $\tan(\phi)$.

Within this work, a maximum longitudinal force of

$$F_{x,\max} = F_{\text{N}} \tan(\phi_{\text{S-S}}) \tag{3.78}$$

and a maximum lateral force of

$$F_{y,\max} = F_{\text{N}} \tan(\phi_{\text{M-S}}) \tag{3.79}$$

are used. This is based on the assumption that the straight grousers lead to form lock with the sand in longitudinal but not in lateral direction.

Pure sliding friction is assumed, meaning that the maximum friction forces from (3.78) and (3.79) can only act opposed to a relative movement between wheel and soil. JANOSI and HANAMOTO's exponential relation of shear length and the resulting shear stress was introduced in Section 3.5.2 and it was described how steady-state assumptions effectively lead to a purely slip-dependent shear stress. Therefore, a slip velocity dependent model is used in this work. The slip velocity is defined as the difference between the wheel rim velocity through its rotation rate and the actual ground-truth velocity of the contact point. To obtain the latter, the translational velocity of the ith contact point from (3.6) is used,

$${}^{\text{c}_i}\boldsymbol{v}_{\text{c}_i|0} = {}^{\text{c}_i}\underline{\boldsymbol{R}}_{\text{s}_i}(\delta_i)\,{}^{\text{s}_i}\boldsymbol{v}_{\text{c}_i|0}(\boldsymbol{x}_{\text{r}}, \boldsymbol{q}_{\text{b}}, \dot{\boldsymbol{q}}_{\text{b}}) \; . \tag{3.80}$$

The steering joint angles $\boldsymbol{q}_{\text{s}}$ were replaced by the steering angle wheel states $\boldsymbol{\delta}$, introduced in Section 3.4,

$$\boldsymbol{\delta} = \boldsymbol{q}_{\text{s}} \; . \tag{3.81}$$

For better readability, the absolute velocity of the ith contact point, expressed in the 's_i'-frame, is abbreviated as

$$\boldsymbol{v}_{\text{c}_i} = {}^{\text{s}_i}\boldsymbol{v}_{\text{c}_i|0} \tag{3.82}$$

in the remainder. Only the longitudinal component of this contact point velocity vector is needed, which can be obtained with the unit vector $\boldsymbol{e}_x$,

$${}^{\text{c}_i}v_{\text{c}_i|0,x} = \boldsymbol{e}_x^{\top}\,{}^{\text{c}_i}\underline{\boldsymbol{R}}_{\text{s}_i}(\delta_i)\boldsymbol{v}_{\text{c}_i} \; . \tag{3.83}$$

To calculate the slip velocity, the longitudinal component of the absolute wheel velocity is subtracted from the wheel rim velocity $\omega_i r$, yielding

$$v_{\text{sl},i} = \omega_i r - {}^{\text{c}_i}v_{\text{c}_i|0,x}\left(\delta_i, \boldsymbol{v}_{\text{c}_i}\left(\boldsymbol{x}_{\text{r}}, \boldsymbol{q}_{\text{b}}, \dot{\boldsymbol{q}}_{\text{b}}\right)\right) \; . \tag{3.84}$$

For friction of rigid objects, the negative signum of the slip velocity is multiplied to the maximum friction force (3.77) or a hyperbolic tangent is used for numerical stability and differentiability. A more stretched hyperbolic tangent function is very similar to the shape of Janosi & Hanamoto's exponential function (3.34) that was shown in Figure 3.4a. It is thus used for the longitudinal and lateral forces, yielding

$$F_{\mathrm{c}_i,x} = F_{\mathrm{c}_i,\mathrm{N}} \tan(\phi_{\text{S-S}}) \tanh(k_{vx} v_{\mathrm{sl},i}) \,, \tag{3.85}$$

$$F_{\mathrm{c}_i,y} = -F_{\mathrm{c}_i,\mathrm{N}} \tan(\phi_{\text{M-S}}) \tanh(k_{vy} {}^{\mathrm{c}_i}v_{\mathrm{c}_i|0,y}) \tag{3.86}$$

with ${}^{\mathrm{c}_i}v_{\mathrm{c}_i|0,y}$ defined for the y-direction analogously to (3.83). The parameters k_{vx} and k_{vy} can be adjusted for anisotropic force development properties of the current wheel-ground contact.

3.8 Combination of Wheel Dynamics and Ground Contact Model

In Section 3.4, the longitudinal force was used as an unknown quantity. A model for this force (3.85) was then developed in Section 3.7 and can now be used in the wheel dynamics (3.29). Furthermore, the concatenation of all six combined wheel-ground dynamics outputs $\boldsymbol{y}_{\mathrm{w}_i}$ needs to match the input to the rover force-torque sum (3.9), which consists of six contact force vectors

$${}^{\mathrm{s}}\boldsymbol{F}_{\mathrm{c}_i} = \begin{bmatrix} {}^{\mathrm{s}}\boldsymbol{F}_{\mathrm{c}_i,\mathrm{T}} \\ {}^{\mathrm{s}}F_{\mathrm{c}_i,\mathrm{N}} \end{bmatrix} . \tag{3.87}$$

The normal forces ${}^{\mathrm{s}}F_{\mathrm{c}_i,\mathrm{N}}$ are estimated through (3.76) and since the z-axes of steering and contact frames are identical for the assumed zero contact angles, ${}^{\mathrm{s}}F_{\mathrm{c}_i,\mathrm{N}} = F_{\mathrm{c}_i,\mathrm{N}}$ holds true. This identity of the two frames, however, is not the case for the x- and y-axes, thus the tangential force in (3.87) was rotated into the 's_i' frame with the steering angle δ_i and the appropriate rotation matrix ${}^{\mathrm{s}_i}\underline{\boldsymbol{R}}_{\mathrm{c}_i}(\delta_i)$.

The combined wheel-ground dynamics system results as

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{\omega}_i \\ \dot{\delta}_i \\ \ddot{\delta}_i \end{bmatrix}}_{\dot{\boldsymbol{x}}_{\mathrm{w}_i}} &= \underbrace{\begin{bmatrix} -\frac{r}{I_{\mathrm{w}_i,y}} F_{\mathrm{c}_i,x} \\ \dot{\delta}_i \\ 0 \end{bmatrix}}_{\boldsymbol{f}_{\mathrm{w}_i}(\boldsymbol{x}_{\mathrm{w}_i}, \boldsymbol{v}_{\mathrm{c}_i})} + \underbrace{\begin{bmatrix} 1/I_{\mathrm{w}_i,y} & 0 \\ 0 & 0 \\ 0 & 1/I_{\mathrm{w}_i,z} \end{bmatrix}}_{\underline{\boldsymbol{B}}_{\mathrm{w}_i}} \underbrace{\begin{bmatrix} T_{\omega,i} \\ T_{\delta,i} \end{bmatrix}}_{\boldsymbol{u}_{\mathrm{w}_i}} \\ \underbrace{{}^{\mathrm{s}_i}\begin{bmatrix} F_{\mathrm{c}_i,x} \\ F_{\mathrm{c}_i,y} \end{bmatrix}}_{\boldsymbol{y}_{\mathrm{w}_i}} &= \underbrace{{}^{\mathrm{s}_i}\underline{\boldsymbol{R}}_{\mathrm{c}_i}(\delta_i) \begin{bmatrix} F_{\mathrm{c}_i,x} \\ F_{\mathrm{c}_i,y} \end{bmatrix}}_{\boldsymbol{h}_{\mathrm{w}_i}(\boldsymbol{x}_{\mathrm{w}_i}, \boldsymbol{v}_{\mathrm{c}_i})} \end{aligned} \tag{3.88}$$

with

$$F_{\mathrm{c}_i,x} = F_{\mathrm{c}_i,\mathrm{N}} \tan(\phi_{\text{S-S}}) \tanh\left(k_{vx}\left(\omega_i r - \boldsymbol{e}_x^{\top\,\mathrm{c}_i}\underline{\boldsymbol{R}}_{\mathrm{s}_i}(\delta_i)\boldsymbol{v}_{\mathrm{c}_i}(\boldsymbol{x}_{\mathrm{r}}, \boldsymbol{q}_{\mathrm{b}}, \dot{\boldsymbol{q}}_{\mathrm{b}})\right)\right) \,, \tag{3.89}$$

$$F_{\mathrm{c}_i,y} = -F_{\mathrm{c}_i,\mathrm{N}} \tan(\phi_{\text{M-S}}) \tanh\left(k_{vy}\boldsymbol{e}_y^{\top\,\mathrm{c}_i}\underline{\boldsymbol{R}}_{\mathrm{s}_i}(\delta_i)\boldsymbol{v}_{\mathrm{c}_i}(\boldsymbol{x}_{\mathrm{r}}, \boldsymbol{q}_{\mathrm{b}}, \dot{\boldsymbol{q}}_{\mathrm{b}})\right) \,. \tag{3.90}$$

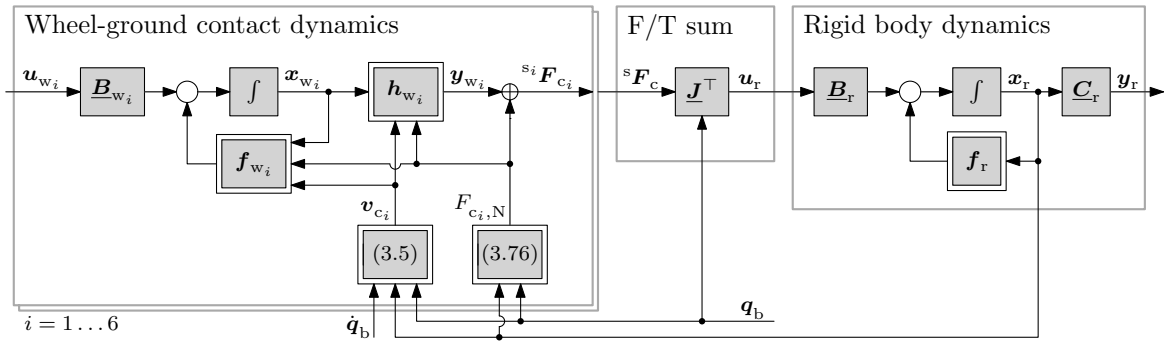


Figure 3.11: Block diagram of the full rover model.

The outputs of the system $\boldsymbol{y}_{\mathrm{w}_i}$ thus match the tangential forces of (3.87) and, together with the normal force estimation from (3.76), they can be used with (3.31) in the force-torque equilibrium (3.9).

3.9 Summary of the Model Development

Three main model blocks were derived in this chapter, an overview of which is shown in Figure 3.11. The actual manipulated variables are steering and drive torques, which form the input vector $\boldsymbol{u}_{\mathrm{w}_i}$ to the wheel-ground dynamics. Through its nonlinear dynamics (3.88), tangential contact forces $\boldsymbol{y}_{\mathrm{w}_i}$ in the non-steered frames of each wheel are computed. Therefore, the velocity of each contact point $\boldsymbol{v}_{\mathrm{c}_i}$ is calculated through (3.5), which requires the rover-level velocity $\boldsymbol{x}_{\mathrm{r}}$ as well as the bogie angles $\boldsymbol{q}_{\mathrm{b}}$ and their derivatives $\dot{\boldsymbol{q}}_{\mathrm{b}}$. For obtaining the third component of the 3-dimensional contact forces, the normal force $F_{\mathrm{c}_i,\mathrm{N}}$ is estimated through the quasi-static assumption with the system of equations (3.76). The full vector of contact forces ${}^{\mathrm{s}}\boldsymbol{F}_{\mathrm{c}}$ results by wheel-wise concatenation (symbol $\oplus$) of the wheel-ground dynamics output $\boldsymbol{y}_{\mathrm{w}_i}$ with the normal force $F_{\mathrm{c}_i,\mathrm{N}}$ and their subsequent concatenation for the six contact points as in (3.31). To calculate the rover CoG forces and torques $\boldsymbol{u}_{\mathrm{r}}$, the concatenated contact forces are multiplied with the transposed Jacobian $\underline{\boldsymbol{J}}^{\top}$, see (3.9). For the computation of the rover motion, the forces and torques $\boldsymbol{u}_{\mathrm{r}}$ are finally fed into the nonlinear rigid body dynamics (3.26).

4 Controller Synthesis

The goal of the overall rover chassis controller is to compute the manipulated variables of the actuators, such that a desired rover movement is achieved. Six drive and six steering torques

$$\boldsymbol{u}_{\mathrm{w}_i} = \begin{bmatrix} T_{\omega_i} \\ T_{\delta_i} \end{bmatrix}, \quad i = 1 \dots 6 \tag{4.1}$$

form the vector of manipulated variables, while the desired rover movement is described by the trajectory of the control variables longitudinal, lateral and yaw velocity,

$$\boldsymbol{y}_{\mathrm{r}}^{*} = \begin{bmatrix} v_{\mathrm{r},x}^{*} \\ v_{\mathrm{r},y}^{*} \\ \omega_{\mathrm{r},z}^{*} \end{bmatrix}. \tag{4.2}$$

Before the controllers can be synthesized, the developed models from Chapter 3 are modified in Section 4.1 to achieve necessary properties of the employed controller structure. A feedback linearization of the nonlinear rigid body dynamics system is designed as system level sub-controller in Section 4.2 to compute the virtual input trajectory. The effectors are the individual wheels including their steering and drive actuation. The actual control allocation therefore computes the individual wheel force trajectories $\boldsymbol{y}_{\mathrm{w}}^{*}$, under consideration of the rover kinematics, and is derived in Section 4.3. To follow the wheel force trajectories, an underlying effector controller is designed with a dynamic extension and another feedback linearization in Section 4.4. Due to the effector controller requiring derivatives of the wheel force trajectories, these derivatives are deduced in Section 4.5, before giving an overview of the full controller in Section 4.6.

4.1 Model Analysis and Modification for the Synthesis

As described in Section 2.7, it is beneficial to use a structure of three sub-controllers for the control of an over-actuated vehicle. In the upper block of Figure 4.1 the corresponding model structure is depicted with the two dynamic model blocks *system level dynamics* and *effector dynamics* as well as the *static allocation* itself. The allocation model is a linear, static equation of the type (2.63) for the rover and can be written with an allocation matrix $\underline{\boldsymbol{G}}_{\mathrm{a}}$ that relates the allocation input $\boldsymbol{u}_{\mathrm{a}}$ to the virtual input $\boldsymbol{y}_{\mathrm{a}}$ as

$$\boldsymbol{y}_{\mathrm{a}} = \underline{\boldsymbol{G}}_{\mathrm{a}} \boldsymbol{u}_{\mathrm{a}}. \tag{4.3}$$

Therein the index 'a' marks the allocation sub-model and the allocation matrix $\underline{\boldsymbol{G}}_{\mathrm{a}}$ has more columns than rows, representing the over-actuation of the system. As indicated in the

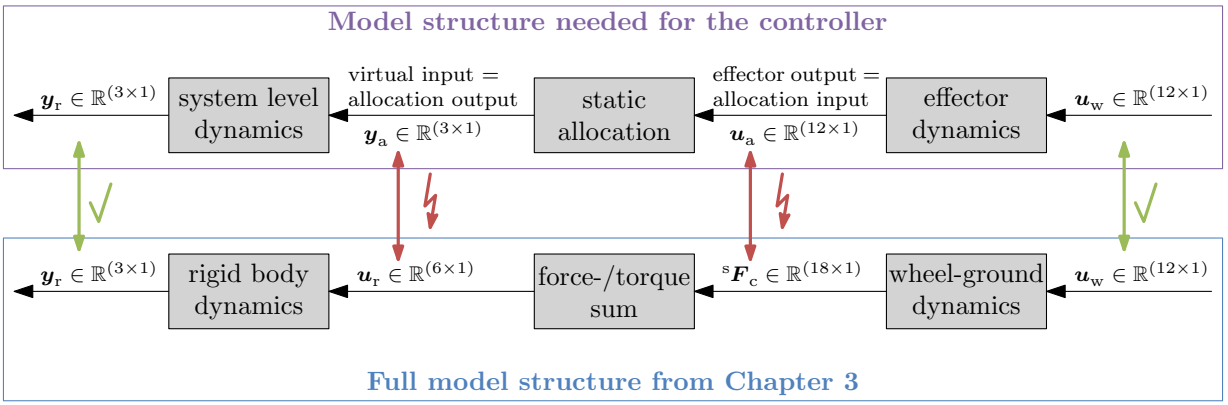


Figure 4.1: Illustration of the discrepancies between the model structure from Chapter 3 and the structure that is needed for the desired controller.

structure in Figure 4.1, the virtual inputs are the outputs of the allocation $\boldsymbol{y}_\mathrm{a}$ and the effector outputs represent the allocation inputs $\boldsymbol{u}_\mathrm{a}$. For the cascaded controller that is designed based on this model structure, each sub-controller is only valid if the underlying sub-controllers are able to achieve the desired trajectory. Additionally, the dynamic sub-models should have the same number of in- and outputs such that there is a unique solution and the over-actuation can be handled exclusively in the allocation. Three specific requirements can be derived from these considerations:

Req 1 The virtual input vector $\boldsymbol{y}_\mathrm{a}$ must be of the same size as the output vector $\boldsymbol{y}_\mathrm{r}$ and must yield an output controllable rigid body dynamics system, see Section 2.6.

Req 2 The virtual input must be achievable by the subsequent sub-controllers, i.e. allocation and effector controller. For the allocation equation (4.3) this achievability implies that the matrix $\underline{\boldsymbol{G}}_\mathrm{a}$ must have full rank, see Section 2.7.

Req 3 Analogously to (Req 1), the effector output $\boldsymbol{u}_\mathrm{a}$ must be of size 12 according to the six drive and six steering torques that are contained in $\boldsymbol{u}_\mathrm{w}$. The effector output, which is also the allocation input, must be achievable by the effector controller. Therefore, the effector sub-model must be output controllable, just like the rigid body dynamics.

The bottom block in Figure 4.1 shows the model structure from Chapter 3 in a simplified form of Figure 3.11. It can be seen that the wheel-ground dynamics output ${}^\mathrm{s}\boldsymbol{F}_\mathrm{c}$ consists of 18 elements, three for each of the six wheels, compared to the desired size of 12 according to (Req 3) from above. However, the model overview diagram in Figure 3.11 shows that the full 3-dimensional force vector is composed of the two tangential forces and one normal force per effector. As described in Section 3.8, the tangential forces are influenced by the wheel-ground dynamics, while the normal forces are not. In fact, the tangential forces are the output of the wheel-ground dynamics (3.88), which has the same number of in- and outputs. Thereby, the first requirement from (Req 3) is fulfilled by taking only the tangential forces

as the allocation input/effector outputs,

$$\boldsymbol{u}_{\mathrm{a}} = \boldsymbol{y}_{\mathrm{w}} = {}^{\mathrm{s}}\boldsymbol{F}_{\mathrm{c,T}} \,. \tag{4.4}$$

For the assessment of the second (Req 3), i.e. the output controllability, the wheel-ground dynamics (3.88) is rewritten with their explicit state-dependencies,

$$\dot{\boldsymbol{x}}_{\mathrm{w}_i} = \begin{bmatrix} a_1(\boldsymbol{x}) \\ x_{\mathrm{w}_i,3} \\ 0 \end{bmatrix} + \begin{bmatrix} 1/I_{\mathrm{w}_i,y} & 0 \\ 0 & 0 \\ 0 & 1/I_{\mathrm{w}_i,z} \end{bmatrix} \begin{bmatrix} u_{\mathrm{w}_i,1} \\ u_{\mathrm{w}_i,2} \end{bmatrix}. \tag{4.5}$$

Comparing this equation with (2.59) shows that it is in the nonlinear canonical form. Applying Lemma 3 to the wheel ground dynamics (4.5), the entries (1,1) and (3,2) must be non-zero for complete state controllability. These two entries are constant non-zero, hence the condition is fulfilled. However, for the controller framework, *output controllability* was required in (Req 3) above. For a linear system, output controllability can be concluded from state controllability if the output matrix has full rank, see Section 2.6. The output equation of (3.88) with (3.89) and (3.90) is nonlinear in the states $\boldsymbol{x}_{\mathrm{w}_i}$, therefore a conclusion cannot easily be drawn. Its linearization around $\boldsymbol{x}_{\mathrm{w}_i} = \boldsymbol{0}$ has full rank, which means that the system is locally output controllable around the origin. However, the states only appear as arguments of hyperbolic tangent, cosine and sine functions, which are all limited to values between -1 and 1. Therefore, the outputs of the wheel ground dynamics are at least limited by the maximum forces (3.78) and (3.79) and therefore not completely output controllable. This is sensible physically, since the maximum forces cannot be exceeded, irrespectively of the wheel rate or the steering angle. For future considerations of limits of the system, the controllability limit of the wheel-ground dynamics should be investigated in more detail and taken into account.

The allocation equation (3.9) needs to be split into the normal and tangential forces as well, since only the latter was chosen as effector output in (4.4). This separation reads

$$\begin{bmatrix} \boldsymbol{F}_{\mathrm{r}} \\ \boldsymbol{T}_{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_{\mathrm{r,N}} \\ \boldsymbol{T}_{\mathrm{r,N}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{F}_{\mathrm{r,T}} \\ \boldsymbol{T}_{\mathrm{r,T}} \end{bmatrix} = \underbrace{\underline{\boldsymbol{J}}_{F,\mathrm{N}}^{\top}\, {}^{\mathrm{s}}\boldsymbol{F}_{\mathrm{c,N}}}_{\text{non-controllable}} + \underbrace{\underline{\boldsymbol{J}}_{F,\mathrm{T}}^{\top}\, {}^{\mathrm{s}}\boldsymbol{F}_{\mathrm{c,T}}}_{\text{controllable}} \,. \tag{4.6}$$

The tangential forces of the last term, labeled with *controllable*, are the chosen 12 effector model outputs $\boldsymbol{y}_{\mathrm{w}}$ from (4.4). The first term of (4.6), containing the normal forces, cannot actively be controlled and is therefore handled separately, which will be detailed later. A similar separation of controllable and non-controllable forces has been done by Waldron and Hubert [WH00] for longitudinal and lateral forces for a case where the latter are non-controllable.

It would seem consequent now to choose the $\underline{\boldsymbol{J}}_{F,\mathrm{T}}^{\top}$-matrix as the allocation matrix $\underline{\boldsymbol{G}}_{\mathrm{a}}$ and the (6×1)-vector $[\boldsymbol{F}_{\mathrm{r}}, \boldsymbol{T}_{\mathrm{r}}]$ minus the normal force term in (4.6) as virtual inputs $\boldsymbol{y}_{\mathrm{a}}$. However, analyzing the (6×12) bogie-angle dependent matrix $\underline{\boldsymbol{J}}_{F,\mathrm{T}}^{\top}(\boldsymbol{q}_{\mathrm{b}})$, it shows rank deficiency for certain bogie configurations. The nominal configuration, $\boldsymbol{q}_{\mathrm{b}} = \boldsymbol{0}$, for example, leads to the

tangential forces purely acting in the x-y-plane of the 'r'-frame. This leads to a rank of the transposed Jacobian of rank $\left(\underline{\boldsymbol{J}}_{F,\mathrm{T}}^{\top}\right) = 3$, i.e. three of the six rows are linearly dependent and only the forces in x- and y-direction and the torque around the vertical z-axis can be controlled. This violates requirement (Req 2) from above, however, the x- and y-force and the yaw torque

$$\begin{bmatrix} F_{\mathrm{r},x} \\ F_{\mathrm{r},y} \\ T_{\mathrm{r},z} \end{bmatrix} , \tag{4.7}$$

form a subset of the rover force-torque vector that turns out to be realizable for the desired operational range of all bogie angles

$$-90^\circ < q_{\mathrm{b}_j} < 90^\circ \quad \forall j = 1 \dots 3 , \tag{4.8}$$

i.e. the associated allocation matrix has full rank within this bogie angle range. The first part of (Req 1) from above is also fulfilled since (4.7) has size three, the same as the rover velocity trajectory $\boldsymbol{y}_{\mathrm{r}}^{*}$ from (4.2). Item Req 1 states that the virtual inputs must not only be of the same size as the rover output but also lead to an output controllable rover dynamics. Looking at the rigid body dynamics (3.26), the block diagonal input matrix $\underline{\boldsymbol{B}}_{\mathrm{r}}$ consists of the inverse mass and inertia matrices. The former is a diagonal matrix and the inertia matrix typically has diagonal entries that are several orders of magnitude higher than its non-diagonal entries for a rover. Hence the rover translational and rotational velocities are mainly influenced by their respective force or torque, i.e. the proposed virtual control input (4.7) has the major impact on the desired control variables (4.2). As described in Section 2.6, there is no general method to proof controllability of a nonlinear system that is not represented in the nonlinear controllable canonical form (2.59). Since the rigid body dynamics is not in this form, it is linearized with the dynamics from (3.26), the input candidate (4.7), the respective columns of the input matrix $\underline{\boldsymbol{B}}_{\mathrm{r}}$ and the output vector $\boldsymbol{y}_{\mathrm{r}}$. The linearization is done around $\boldsymbol{x}_{\mathrm{r}} = \boldsymbol{0}$ to make use of Lemma 4. The rank of the linearized rigid body dynamics' output controllability matrix equals the number of system outputs,

$$\operatorname{rank}(\underline{\boldsymbol{S}}_{\mathrm{oc,r}}) = 3 , \tag{4.9}$$

hence the linearized system is output controllable. For the original nonlinear rigid body dynamics, this implies at least local output controllability around $\boldsymbol{x}_{\mathrm{r}} = \boldsymbol{0}$.

The vector (4.7) is thus chosen as virtual input and allocation output

$$\boldsymbol{y}_{\mathrm{a}} = \tilde{\boldsymbol{u}}_{\mathrm{r}} = \begin{bmatrix} F_{\mathrm{r},x} \\ F_{\mathrm{r},y} \\ T_{\mathrm{r},z} \end{bmatrix} . \tag{4.10}$$

with the associated allocation matrix

$$\underline{\boldsymbol{G}}_{\mathrm{a}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \underline{\boldsymbol{J}}_{F,\mathrm{T}}^{\top}(\boldsymbol{q}_{\mathrm{b}}) . \tag{4.11}$$

The force-torque balance (4.6) can now be rewritten again as

$$\begin{bmatrix} \boldsymbol{F}_{\mathrm{r}} \\ \boldsymbol{T}_{\mathrm{r}} \end{bmatrix} = \underbrace{\underline{\boldsymbol{J}}_{F,\mathrm{N}}^{\top}\,{}^{\mathrm{s}}\boldsymbol{F}_{\mathrm{c,N}}}_{\text{non-controllable}} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\underline{\boldsymbol{G}}_{\mathrm{a}}\,{}^{\mathrm{s}}\boldsymbol{F}_{\mathrm{c,T}}}_{=\tilde{\boldsymbol{u}}_{\mathrm{r}}} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \underline{\boldsymbol{J}}_{F,\mathrm{T}}^{\top}\,{}^{\mathrm{s}}\boldsymbol{F}_{\mathrm{c,T}}}_{= \begin{bmatrix} F_{\mathrm{r},z} \\ T_{\mathrm{r},x} \\ T_{\mathrm{r},y} \end{bmatrix}, \text{ non-controllable}} . \tag{4.12}$$

It can be seen that, with the choice of the allocation equation

$$\tilde{\boldsymbol{u}}_{\mathrm{r}} = \underline{\boldsymbol{G}}_{\mathrm{a}}\,{}^{\mathrm{s}}\boldsymbol{F}_{\mathrm{c,T}} , \tag{4.13}$$

only the middle term of the force-torque balance (4.12) is considered. The first term is non-controllable due to the normal forces not being actively adjustable with the available actuators for driving and steering. They are, however, estimated in Section 3.7.2, using dependencies on measurable states and bogie angles only. The normal force term is thus added to the nonlinear function $\boldsymbol{f}_{\mathrm{r}}$ of the rigid body dynamics (3.26), which leads to the modified nonlinear function

$$\tilde{\boldsymbol{f}}_{\mathrm{r}} = \boldsymbol{f}_{\mathrm{r}} + \underline{\boldsymbol{B}}_{\mathrm{r}}\underline{\boldsymbol{J}}_{F,\mathrm{N}}^{\top}(\boldsymbol{q}_{\mathrm{b}})\boldsymbol{F}_{\mathrm{c,N}}(\boldsymbol{q}_{\mathrm{b}}, \boldsymbol{\theta}_{\mathrm{r}}) . \tag{4.14}$$

Even though the last term in (4.12) depends on the controllable tangential contact forces, it was shown above that it can become non-controllable for some bogie configurations due to a rank deficiency of the associated rows of the Jacobian. This term can therefore not be handled in the allocation together with the middle term of (4.12), but must be added to the rigid body dynamics as a disturbance.

These modifications lead to the new synthesis model of the rigid body dynamics

$$\begin{aligned} \dot{\boldsymbol{x}}_{\mathrm{r}} &= \tilde{\boldsymbol{f}}_{\mathrm{r}}(\boldsymbol{x}_{\mathrm{r}}, \boldsymbol{q}_{\mathrm{b}}) + \underline{\widetilde{\boldsymbol{B}}}_{\mathrm{r}}\tilde{\boldsymbol{u}}_{\mathrm{r}} + \underline{\boldsymbol{E}}_{\mathrm{r}}\boldsymbol{z}_{\mathrm{r}F} \\ \boldsymbol{y}_{\mathrm{r}} &= \underline{\boldsymbol{C}}_{\mathrm{r}}\boldsymbol{x}_{\mathrm{r}} \end{aligned} \tag{4.15}$$

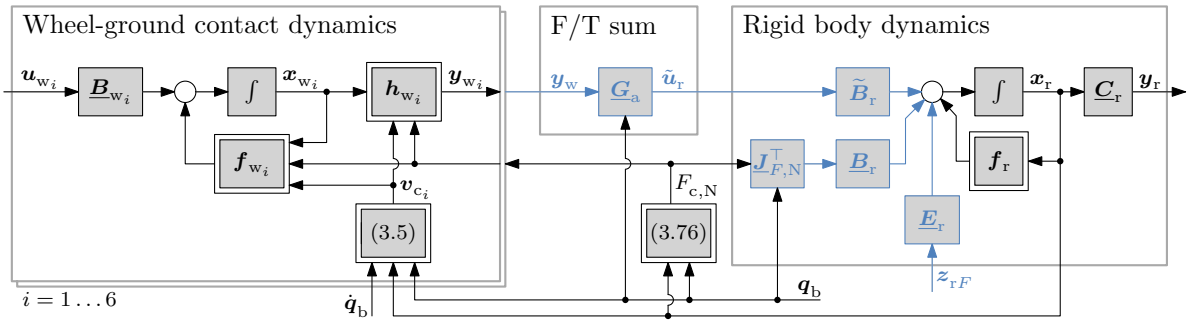


Figure 4.2: Modified model overview for the controller synthesis, where the modified or added parts are colored blue.

with

$$
\begin{aligned}
\boldsymbol{x}_\mathrm{r} &= \begin{bmatrix} {}^\mathrm{r}\boldsymbol{v}_{\mathrm{r}|0} \\ {}^\mathrm{r}\boldsymbol{\omega}_{\mathrm{r}|0} \\ \boldsymbol{\theta}_\mathrm{r} \end{bmatrix}, \tilde{\boldsymbol{u}}_\mathrm{r} = \begin{bmatrix} F_{\mathrm{r},x} \\ F_{\mathrm{r},y} \\ T_{\mathrm{r},z} \end{bmatrix}, \boldsymbol{z}_{\mathrm{r}F} = \begin{bmatrix} F_{\mathrm{r},z} \\ T_{\mathrm{r},x} \\ T_{\mathrm{r},y} \end{bmatrix}, \\
\tilde{\boldsymbol{f}}_\mathrm{r} &= \begin{bmatrix} -\underline{\boldsymbol{S}}({}^\mathrm{r}\boldsymbol{\omega}_{\mathrm{r}|0}){}^\mathrm{r}\boldsymbol{v}_{\mathrm{r}|0} + {}^\mathrm{r}\underline{\boldsymbol{R}}_0(\boldsymbol{\theta}_\mathrm{r}){}^0\boldsymbol{a}_\mathrm{g} \\ \boldsymbol{0} \\ \underline{\boldsymbol{U}}^{-1}(\boldsymbol{\theta}_\mathrm{r}){}^\mathrm{r}\boldsymbol{\omega}_{\mathrm{r}|0} \end{bmatrix} + \underline{\boldsymbol{B}}_\mathrm{r}\underline{\boldsymbol{J}}_{F,\mathrm{N}}^\top(\boldsymbol{q}_\mathrm{b})\boldsymbol{F}_{\mathrm{c,N}}(\boldsymbol{q}_\mathrm{b}, \boldsymbol{\theta}_\mathrm{r}), \\
\underline{\boldsymbol{B}}_\mathrm{r} &= \begin{bmatrix} \underline{\boldsymbol{M}}^{-1} & \underline{\boldsymbol{0}} \\ \underline{\boldsymbol{0}} & \underline{\boldsymbol{\Theta}}^{-1} \\ \underline{\boldsymbol{0}} & \underline{\boldsymbol{0}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_{\mathrm{r},1} & \dots & \boldsymbol{b}_{\mathrm{r},6} \end{bmatrix}, \\
\underline{\widetilde{\boldsymbol{B}}}_\mathrm{r} &= \begin{bmatrix} \boldsymbol{b}_{\mathrm{r},1} & \boldsymbol{b}_{\mathrm{r},2} & \boldsymbol{b}_{\mathrm{r},6} \end{bmatrix}, \\
\underline{\boldsymbol{E}}_\mathrm{r} &= \begin{bmatrix} \boldsymbol{b}_{\mathrm{r},3} & \boldsymbol{b}_{\mathrm{r},4} & \boldsymbol{b}_{\mathrm{r},5} \end{bmatrix}, \\
\underline{\boldsymbol{C}}_\mathrm{r} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.
\end{aligned}
\tag{4.16}
$$

In Figure 4.2, the modified model structure is depicted. Compared to the original model structure, that was shown in Figure 3.11, only the tangential wheel forces $\boldsymbol{y}_\mathrm{w} = {}^\mathrm{s}\boldsymbol{F}_{\mathrm{c},T}$ are input to the force-torque sum. The normal force term, on the other side, is added to the nonlinear dynamics function as in (4.14). Ultimately, the output of the force-torque sum has changed as well: $\tilde{\boldsymbol{u}}_\mathrm{r}$ is only a subset of the full force-torque vector $\boldsymbol{u}_\mathrm{r}$ and the remaining force and torques are treated as disturbances $\boldsymbol{z}_{\mathrm{r}F}$ and added to the rigid body dynamics with the disturbance matrix $\underline{\boldsymbol{E}}_\mathrm{r}$.

4.2 Rigid Body Dynamics Feedback Linearization

In this section, a feedback linearization controller is designed for the modified rigid body dynamics model (4.15). After the synthesis of the control law in Section 4.2.1, the stability of the resulting internal dynamics is investigated in Section 4.2.2.

4.2.1 Feedback Linearization Control Law

The modified rigid body dynamics from (4.15) is a system with $n_x = 9$ states, $n_u = 3$ inputs and $n_y = 3$ outputs, which are the first, second and sixth state. This section is based on the general process of a feedback linearization controller synthesis for a MIMO system that was summarized in Section 2.4.

From (4.15) and (4.16), the output derivatives are

$$\dot{\boldsymbol{y}}_\mathrm{r} = \begin{bmatrix} \dot{x}_{\mathrm{r},1} \\ \dot{x}_{\mathrm{r},2} \\ \dot{x}_{\mathrm{r},6} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{f}}_{\mathrm{r},1} \\ \dot{\tilde{f}}_{\mathrm{r},2} \\ \dot{\tilde{f}}_{\mathrm{r},6} \end{bmatrix} + \underline{\boldsymbol{C}}_\mathrm{r} \underline{\widetilde{\boldsymbol{B}}}_\mathrm{r} \tilde{\boldsymbol{u}}_\mathrm{r} \,. \tag{4.17}$$

The disturbances $\boldsymbol{z}_{\mathrm{r}F}$ from (4.15) are not considered in the feedback linearization. Note that, for purely diagonal mass and inertia matrices, $\underline{\boldsymbol{C}}_\mathrm{r} \underline{\boldsymbol{E}}_\mathrm{r} = \underline{\boldsymbol{0}}$ holds true, which means that the disturbances would not appear in the output derivative anyway. For non-zero non-diagonal elements with dominant diagonal entries, there are comparatively small influences of the disturbance on the output derivative (4.17). This small impact is neglected by the feedback linearization controller design. As the disturbances are forces and torques on the rover level, they could be calculated with the contact forces, e.g. of the last timestep through (4.6) and (4.16), and used for a disturbance rejection. Neither the design nor the effect of such a system is investigated here, but the impact on the outputs is expected to be small, due to the comparatively small influence of the disturbance on the output derivative that was described above. Due to the controller using the measured states the effect of the disturbances is, however, considered indirectly.

The inputs appear in the first derivative of each of the outputs, hence the system has a vector relative degree of

$$\boldsymbol{p}_\mathrm{r} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} . \tag{4.18}$$

Due to the constant input and output matrices, the Lie-derivatives in the $\underline{\boldsymbol{V}}$-matrix, see (2.35), simplify, which yields the constant matrix

$$\underline{\boldsymbol{V}}_\mathrm{r} = \underline{\boldsymbol{C}}_\mathrm{r} \underline{\widetilde{\boldsymbol{B}}}_\mathrm{r} = \begin{bmatrix} 1/m & 0 & 0 \\ 0 & 1/m & 0 \\ 0 & 0 & (\Theta_\mathrm{r}^{-1})_{33} \end{bmatrix} \tag{4.19}$$

with $(\Theta_\mathrm{r}^{-1})_{33}$ the $(3,3)$ element of the inverted inertia matrix $\underline{\boldsymbol{\Theta}}_\mathrm{r}^{-1}$.

The new coordinates $\boldsymbol{\xi}_\mathrm{r}$ are trivial to find through (2.40), since, for a relative degree of one, they are simply the output functions themselves,

$$\boldsymbol{\xi}_\mathrm{r} = \begin{bmatrix} x_{\mathrm{r},1} & x_{\mathrm{r},2} & x_{\mathrm{r},6} \end{bmatrix}^\top . \tag{4.20}$$

For asymptotic trajectory tracking, the output trajectories $\boldsymbol{y}_\mathrm{r}^*$ and their first derivatives $\dot{\boldsymbol{y}}_\mathrm{r}^*$ are needed due to the relative degree of one. With the diagonal controller gain matrix

$$\underline{\boldsymbol{k}}_{\mathrm{r},0} = \mathrm{diag}\left(\begin{bmatrix} k_{\mathrm{r},0,1} & k_{\mathrm{r},0,2} & k_{\mathrm{r},0,3} \end{bmatrix} \right) , \tag{4.21}$$

the control law (2.44) simplifies to

$$\tilde{\boldsymbol{u}}_{\mathrm{r,asympt}}^{*} = \underline{\boldsymbol{V}}_{\mathrm{r}}^{-1} \left(-\underline{\boldsymbol{C}}_{\mathrm{r}} \tilde{\boldsymbol{f}}_{\mathrm{r}} + \dot{\boldsymbol{y}}_{\mathrm{r}}^{*} - \underline{\boldsymbol{k}}_{\mathrm{r},0} \left(\boldsymbol{y}_{\mathrm{r}} - \boldsymbol{y}_{\mathrm{r}}^{*} \right) \right) . \tag{4.22}$$

The first term in parentheses, $-\underline{\boldsymbol{C}}_{\mathrm{r}} \tilde{\boldsymbol{f}}_{\mathrm{r}}$, is the $\boldsymbol{w}$-vector from (2.42), where the Lie derivatives result in the simple multiplication of output matrix and $\tilde{\boldsymbol{f}}_{\mathrm{r}}$-vector for a linear output equation. To compensate model errors and unknown disturbances, e.g. non-zero contact angles or ground resistance, an additional integrator term is added to the control law, yielding the final rigid body dynamics control law

$$\tilde{\boldsymbol{u}}_{\mathrm{r}}^{*} = \underline{\boldsymbol{V}}_{\mathrm{r}}^{-1} \left(-\underline{\boldsymbol{C}}_{\mathrm{r}} \tilde{\boldsymbol{f}}_{\mathrm{r}} + \dot{\boldsymbol{y}}_{\mathrm{r}}^{*} - \underline{\boldsymbol{k}}_{\mathrm{r},0} \left(\boldsymbol{y}_{\mathrm{r}} - \boldsymbol{y}_{\mathrm{r}}^{*} \right) - \underline{\boldsymbol{k}}_{\mathrm{r,I}} \int \left(\boldsymbol{y}_{\mathrm{r}} - \boldsymbol{y}_{\mathrm{r}}^{*} \right) \mathrm{d}t \right) . \tag{4.23}$$

Therein $\underline{\boldsymbol{k}}_{\mathrm{r,I}}$ is another diagonal gain matrix for the integrator term of the same dimension and structure as $\underline{\boldsymbol{k}}_{\mathrm{r},0}$.

Due to not considering the limits of the actuators or the soil contact directly, see also the result discussion in Section 5.4, a windup of the integrator in (4.23) is possible. To prevent that, the measured actual wheel states are used in the ground contact model and the allocation (4.13) is used to obtain an estimation of the achieved rover forces $\tilde{\boldsymbol{u}}_{\mathrm{r}}$. The delta between desired and estimated rover forces is computed and can be employed to set the integrator input $(\boldsymbol{y}_{\mathrm{r}} - \boldsymbol{y}_{\mathrm{r}}^{*})$ to zero to prevent a windup if the delta gets too large. This is based on the idea that the respective forces might not always be directly achievable depending on the steering angle configuration or actuator saturation and that this is partly indicated by the described delta between desired and achieved forces. In the simulation examples in Chapter 5, this anti-windup has shown good improvements of the result, especially for desired movements that require a change of the steering angles before being achievable. These types of trajectories lead to large steering rates, which are not feasible due to the factual limits of the actuators that are not considered in this synthesis. Again, the topic of non-considered steering rate limits will be described and discussed in Chapter 5 with its influence on the results.

4.2.2 Stability of the Internal Dynamics

The vector relative degree of the modified rigid body formulation was derived in (4.18) and the sum of its elements is

$$\sum_{i=1}^{3} p_{\mathrm{r},i} = 3 < n_x = 9 \, . \tag{4.24}$$

As presented in Section 2.4, this difference leads to internal dynamics, which cannot be seen in the stabilized closed-loop outputs and therefore their stability needs to be investigated separately. In the present case, where the coordinates $\boldsymbol{\xi}_{\mathrm{r}}$ are simply three of the states $\boldsymbol{x}_{\mathrm{r}}$, the $9 - 3 = 6$ states of the internal dynamics $\boldsymbol{\eta}_{\mathrm{r}}$ immediately result as the remaining states

of $\boldsymbol{x}_\mathrm{r}$:

$$\boldsymbol{\eta}_\mathrm{r} = \begin{bmatrix} x_{\mathrm{r},3} \\ x_{\mathrm{r},4} \\ x_{\mathrm{r},5} \\ x_{\mathrm{r},7} \\ x_{\mathrm{r},8} \\ x_{\mathrm{r},9} \end{bmatrix} = \begin{bmatrix} {}^\mathrm{r}v_{\mathrm{r}|0,z} \\ {}^\mathrm{r}\omega_{\mathrm{r}|0,x} \\ {}^\mathrm{r}\omega_{\mathrm{r}|0,y} \\ \phi \\ \theta \\ \psi \end{bmatrix} \tag{4.25}$$

with their dynamics $\dot{\boldsymbol{\eta}}_\mathrm{r}$ given through the respective rows of (4.15).

Position-dependency of the normal force model. Before zeroing the outputs for the investigation of the zero dynamics, the normal force model needs to be revised. Two different ways to calculate the wheel normal forces were presented in Sections 3.7.1 and 3.7.2. The first of them is penetration-based and thus requires the knowledge of the rover's vertical position and the ground height at each wheel. Due to these quantities being non-measurable, the quasi-static estimation, which only uses the rover's orientation and the bogie angles, is used for the controller synthesis. The internal states vector $\boldsymbol{\eta}$, however, includes the vertical velocity of the rover and its roll and pitch rate, all of which are mainly influenced by the normal forces. Through the quasi-static estimation, the normal forces do not change for a vertical movement of the rover, since the estimation is purely orientation-based. However, the position- and ground dependency of the contact normal forces is physically uncontroversial and must be included in the model in order to stabilize the rover in said directions. Figuratively spoken, a lack of this dependency would lead to unbraked acceleration in the vertical direction, which is obviously neither mathematically stable nor physically reasonable.

The spring-damper normal force model from Section 3.7.1 features this dependency on the rover's vertical position $r_{\mathrm{r}|0,z}$, which changes dynamically with the rover's velocity

$$\dot{r}_{\mathrm{r}|0,z} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} {}^0\underline{\boldsymbol{R}}_\mathrm{r} \, {}^\mathrm{r}\boldsymbol{v}_{\mathrm{r}|0} \, . \tag{4.26}$$

Through (3.67) and (3.68), the spring-damper model also brings a dependency of the normal forces on the rotation rates and RPY angles. The rover's vertical position is added as an additional state to the internal states vector

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_\mathrm{r} \\ \dot{r}_{\mathrm{r}|0,z} \end{bmatrix} = \dot{\tilde{\boldsymbol{\eta}}}_\mathrm{r} = \begin{bmatrix} \underline{\boldsymbol{H}}_\mathrm{r} \tilde{\boldsymbol{f}}_{\mathrm{r,springDamper}}(\boldsymbol{x}_\mathrm{r}, r_{\mathrm{r}|0,z}, \boldsymbol{q}_\mathrm{b}, \boldsymbol{z}_\mathrm{g}, \dot{\boldsymbol{z}}_\mathrm{g}) \\ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} {}^0\underline{\boldsymbol{R}}_\mathrm{r}(\boldsymbol{\theta}_\mathrm{r}) {}^\mathrm{r}\boldsymbol{v}_{\mathrm{r}|0} \end{bmatrix} , \tag{4.27}$$

where $\tilde{\boldsymbol{f}}_{\mathrm{r,springDamper}}$ is the vector $\tilde{\boldsymbol{f}}_\mathrm{r}$ from (4.16) with the spring-damper normal force vector $\boldsymbol{F}_{\mathrm{N,springDamper}}$ from (3.69) instead of the quasi-stationary estimation $\boldsymbol{F}_{\mathrm{c,N}}$ from (3.76). The

matrix

$$\underline{\boldsymbol{H}}_{\mathrm{r}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.28}$$

extracts the six rows of $\tilde{\boldsymbol{f}}_{\mathrm{r,springDamper}}$ that represent the zero dynamics states, i.e. the vertical velocity, roll and pitch rates as well as the RPY angles. The disturbances $\boldsymbol{z}_{\mathrm{r}F}$ have an influence on the internal states through the disturbance matrix $\underline{\boldsymbol{E}}_{\mathrm{r}}$ but the robustness to these disturbances is not investigated here. The respective rows of the input matrix $\underline{\widetilde{\boldsymbol{B}}}_{\mathrm{r}}$ are zero, which is why (4.27) shows no input dependency.

Zeroing the outputs and finding the equilibrium. For the stability analysis of the internal dynamics, see Section 2.4, the outputs $\boldsymbol{y}_{\mathrm{r}}$ and their derivatives are zeroed,

$$\boldsymbol{y}_{\mathrm{r}}^{*} = \dot{\boldsymbol{y}}_{\mathrm{r}}^{*} = \boldsymbol{y}_{\mathrm{r}} = \boldsymbol{0} \ , \tag{4.29}$$

and thereby the transformed states and their derivatives must stay zero as well,

$$\boldsymbol{\xi} = \dot{\boldsymbol{\xi}} = \boldsymbol{0} \ . \tag{4.30}$$

Using (4.30) in (4.27) is called *zeroing the outputs* and leads to the zero dynamics of the closed loop system

$$\dot{\tilde{\boldsymbol{\eta}}}_{\mathrm{r}} = \underbrace{\begin{bmatrix} \underline{\boldsymbol{H}}_{\mathrm{r}} \tilde{\boldsymbol{f}}_{\mathrm{r,springDamper}}(\boldsymbol{\xi}_{\mathrm{r}} = \boldsymbol{0}, \tilde{\boldsymbol{\eta}}_{\mathrm{r}}, \boldsymbol{q}_{\mathrm{b}}, \boldsymbol{z}_{\mathrm{g}}, \dot{\boldsymbol{z}}_{\mathrm{g}}) \\ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} {}^{0}\underline{\boldsymbol{R}}_{\mathrm{r}}(\boldsymbol{\theta}_{\mathrm{r}}) \begin{bmatrix} 0 \\ 0 \\ {}^{\mathrm{r}}v_{\mathrm{r}\vert 0,z} \end{bmatrix} \end{bmatrix}}_{\boldsymbol{f}_{\tilde{\eta}}} . \tag{4.31}$$

For stability analysis according to Section 2.5, an equilibrium at $\tilde{\boldsymbol{\eta}}_{\mathrm{r},0} = \boldsymbol{0}$ is investigated without loss of generality, since any equilibrium can be shifted to zero. Due to the position-dependent normal forces, the soil height $\boldsymbol{z}_{\mathrm{g}}$ influences the equilibrium position and orientation of the rover and thus the equilibrium state $\tilde{\boldsymbol{\eta}}_{\mathrm{r},0}$ of the internal dynamics. In the present case, the soil height disturbance is set such that the equilibrium of the zero dynamics is in the origin by setting

$$\begin{aligned} \tilde{\boldsymbol{\eta}}_{\mathrm{r},0} &= \boldsymbol{0} \ , \\ \boldsymbol{q}_{\mathrm{b},0} &= \boldsymbol{0} \ , \\ \dot{\boldsymbol{z}}_{\mathrm{g},0} &= \boldsymbol{0} \ , \end{aligned} \tag{4.32}$$

and solving the equilibrium condition

$$\dot{\tilde{\boldsymbol{\eta}}}_{\mathrm{r},0} = \boldsymbol{0} \tag{4.33}$$

for the soil heights $\boldsymbol{z}_{\mathrm{g},0}$. Depending on the soil contact stiffness parameter k_s from (3.69), these soil heights for the equilibrium in $\tilde{\boldsymbol{\eta}}_{\mathrm{r},0} = \mathbf{0}$ change. Due to the specific kinematics of the ExoMars BB2 rover, the soil heights for an equilibrium in the origin must always be slightly higher at the rear wheels. As an example for a spring constant of $k_\mathrm{s} = 10^5$, solving (4.33) with (4.32) yields the soil heights

$$\boldsymbol{z}_{\mathrm{g},0} = \begin{bmatrix} -0.5944\,\mathrm{m} \\ -0.5944\,\mathrm{m} \\ -0.5943\,\mathrm{m} \\ -0.5943\,\mathrm{m} \\ -0.5944\,\mathrm{m} \\ -0.5944\,\mathrm{m} \end{bmatrix} . \tag{4.34}$$

Stability analysis of the equilibrium. The stability analysis for a nonlinear system according to LYAPUNOV was introduced in Section 2.5. The sum of squares of the internal states

$$V_2(\tilde{\boldsymbol{\eta}}_\mathrm{r}) = \left(\|\tilde{\boldsymbol{\eta}}_\mathrm{r}\|_2\right)^2 \tag{4.35}$$

was investigated as a Lyapunov function candidate. Its Jacobian is, however, not negative definite or negative semidefinite, thus no conclusion can be drawn from this analysis. The dynamics equation of $\tilde{\boldsymbol{\eta}}_\mathrm{r}$ consists of many additive terms that oftentimes include a product of two or more trigonometric functions of the RPY angle states and bogie angle disturbances. A Lyapunov function could therefore not be constructed by hand either.

The indirect Lyapunov method can be used in cases, where a Lyapunov function for the direct method cannot be found. Linearization of (4.31) yields

$$\dot{\tilde{\boldsymbol{\eta}}}_{\mathrm{r,lin}} = \underbrace{\left.\frac{\partial \boldsymbol{f}_{\tilde{\eta}}}{\partial \tilde{\boldsymbol{\eta}}_\mathrm{r}}\right|_{\tilde{\boldsymbol{\eta}}_\mathrm{r}=\mathbf{0}}}_{\underline{\boldsymbol{A}}_\mathrm{lin}(\boldsymbol{q}_\mathrm{b})} \tilde{\boldsymbol{\eta}}_\mathrm{r} . \tag{4.36}$$

and the eigenvalues of the linearization matrix $\underline{\boldsymbol{A}}_\mathrm{lin}$ can be investigated. A numerical analysis has shown that of the seven eigenvalues of $\underline{\boldsymbol{A}}_\mathrm{lin}$, one is exactly zero and the other six have negative real parts for a bogie angle range of

$$-77^\circ < q_{\mathrm{b}_j} < 77^\circ \quad j = 1 \ldots 3 . \tag{4.37}$$

Outside of this bogie angle range, one or more real parts of the eigenvalues become positive and thus the origin becomes unstable. As an example, the real parts of the eigenvalues for $\boldsymbol{q}_\mathrm{b} = \mathbf{0}$ are

$$\mathrm{Re}\left(\boldsymbol{\lambda}(\boldsymbol{q}_\mathrm{b} = \mathbf{0})\right) = \begin{bmatrix} 0 \\ -16.5295 \\ -16.5646 \\ -16.7076 \\ -294.7720 \\ -347.7298 \\ -363.9478 \end{bmatrix} . \tag{4.38}$$

At this point, no conclusion can be drawn for the stability of the origin due to the one real part of zero. A closer look at the linearization matrix $\underline{\boldsymbol{A}}_{\text{lin}}$ reveals that its 6th row and column are both zero. Hence, the yaw angle zero dynamics, linearized around the origin, are

$$\dot{\psi}_{\text{lin}} = 0 \; . \tag{4.39}$$

While the yaw angle is thus not stabilized in its equilibrium, it also does not grow due to its derivative being zero. This is an expected behavior for mechanical systems that are not subject to a position-based force such as a rotational spring. Figuratively spoken, there is no torque that keeps the yaw angle of the rover at any specific value, in contrast to the vertical rover position and its roll and pitch angle, which are stabilized by the spring-damper normal forces. The yaw rate, is stabilized through the feedback linearization controller, since it is part of the coordinates $\boldsymbol{\xi}$ and not of the internal dynamics. As a result, the yaw rate is stable while the yaw angle can be arbitrary. Due to the zero column in $\underline{\boldsymbol{A}}_{\text{lin}}$, this arbitrary yaw angle has no influence on any of the internal states. By removing the yaw angle from (4.36), the zero row and column vanish from $\underline{\boldsymbol{A}}_{\text{lin}}$ and so does the zero eigenvalue. This leaves only the six eigenvalues with negative real parts for the bogie range (4.37).

Figure 4.3 shows the variation of the eigenvalues of the linearization matrix to changing states and bogie angles. Only the respective state or disturbance is varied for each plot, while all others are kept at zero. This is no stability proof for the full domain but gives a good impression of the sensitivity of the eigenvalues within the operational range. The vertical velocity as well as roll and pitch rate do not change the eigenvalues at all, as can be seen in the left three plots. When it comes to the roll and pitch angles, all eigenvalues stay in the left halfplane for angles between $\pm 45°$. The variation of the yaw angle or one bogie angle at a time cannot destabilize the linearization matrix at all between $\pm 90°$, while a full variation has shown that the limits from (4.37) apply if all bogie angles are changed at the same time. All these limits are, larger than the envisioned operational range anyways.

In conclusion, a rigorous stability proof of the nonlinear zero dynamics could not be pursued. A linear analysis of the equilibrium has shown that the yaw angle derivative is zero and the yaw angle does not influence any of the other zero dynamics states' derivatives. This means that the yaw rate is stable, but the specific yaw angle at which the rover is stabilized is not necessarily in the origin, however, it also does not influence the other states in the equilibrium. The yaw angle leads to one eigenvalue with a real part of zero, while all other eigenvalues have negative real parts at the equilibrium position. A sensitivity analysis has shown that all but the one zero eigenvalue stay in the negative halfplane for the operational range of the rover. Physical considerations finally suggest that the rover will, indeed, not fall through the ground, tip over or accelerate uncontrollably in the pitch and roll directions within the operational range of orientation and bogie angles. Due to a good match of this consideration with the above stability analysis of the linear system, stability of the zero dynamics is concluded.

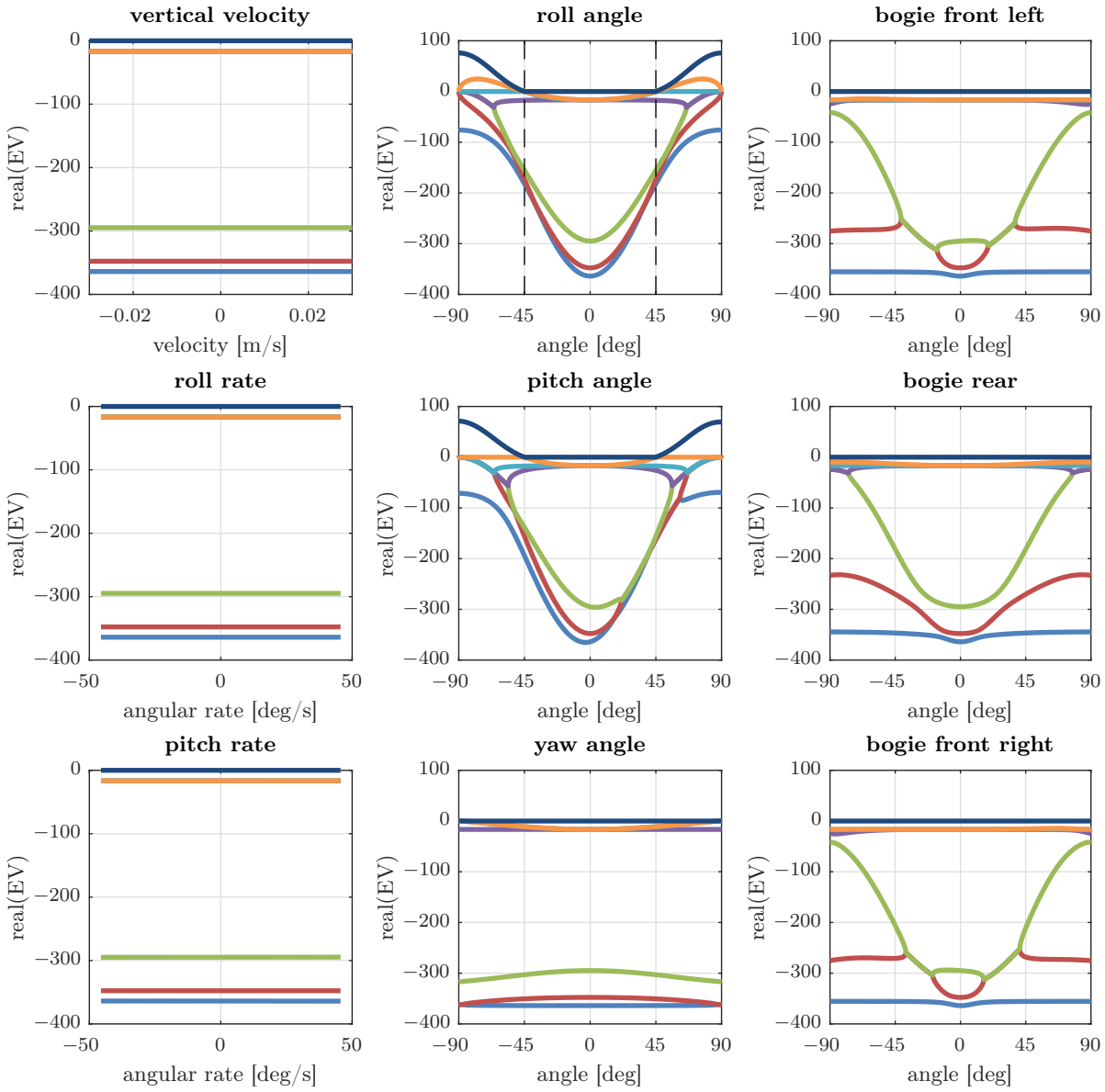


Figure 4.3: Real parts of the linearized zero dynamics' eigenvalues (y-axes) for variation of one state/disturbance at a time (x-axes). Each color represents one of the seven eigenvalues.

4.3 Wheel Force Allocation and Optimization of the Available Degrees of Freedom

The allocation module is responsible for mapping the virtual input trajectories of the system level controller, $\tilde{\boldsymbol{u}}_{\mathrm{r}}^{*}$, to the output trajectories of the effector sub-modules $\boldsymbol{y}_{\mathrm{w}}^{*}$, see Figure 4.2. In this work, the allocation model is the partial force-torque balance of the rover system, which relates the rover-level forward and sideways forces as well as the yaw torque to the

tangential wheel-ground contact forces. It was derived in Section 4.1 and is repeated here:

$$\underbrace{\tilde{\boldsymbol{u}}_\mathrm{r}}_{(3\times1)} = \underbrace{\underline{\boldsymbol{G}}_\mathrm{a}(\boldsymbol{q}_\mathrm{b})}_{(3\times12)} \underbrace{{}^\mathrm{s}\boldsymbol{F}_{\mathrm{c},T}}_{(12\times1)} , \tag{4.40}$$

with the virtual inputs $\tilde{\boldsymbol{u}}_\mathrm{r}$ from (4.10) and the allocation matrix $\underline{\boldsymbol{G}}_\mathrm{a}$ from (4.11). The effector forces

$${}^\mathrm{s}\boldsymbol{F}_{\mathrm{c,T}} = \begin{bmatrix} {}^{\mathrm{s}_1}\boldsymbol{F}_{\mathrm{c}_1,\mathrm{T}} \\ \vdots \\ {}^{\mathrm{s}_6}\boldsymbol{F}_{\mathrm{c}_6,\mathrm{T}} \end{bmatrix} \quad \text{with} \quad {}^{\mathrm{s}_i}\boldsymbol{F}_{\mathrm{c}_i,\mathrm{T}} = \begin{bmatrix} {}^{\mathrm{s}_i}F_{\mathrm{c}_i,x} \\ {}^{\mathrm{s}_i}F_{\mathrm{c}_i,y} \end{bmatrix}, \quad i = 1\ldots6 \tag{4.41}$$

are the tangential wheel contact forces at the six contact points. Remember that the contact forces are expressed in their respective non-steered 's_i'-frame in order to separate the dynamic steering angle state from the allocation. The steering angles are treated in the underlying effector controller, see Section 4.4, and the contact forces are rotated into the non-steered 's_i' frame in the output equation (3.88).

The allocation equation (4.40) is a static, linear, underdetermined system of equations in the form of (2.63). For the six wheel-ground contact points of the ExoMars BB2, $\underline{\boldsymbol{G}}_\mathrm{a}$ has three rows and 12 columns, which means that the system of equations (4.40) has a degree of redundancy of nine. A simple inversion of (4.40), to obtain the contact force trajectories ${}^\mathrm{s}\boldsymbol{F}^*_{\mathrm{c},T}$ from the virtual input trajectory $\tilde{\boldsymbol{u}}^*_\mathrm{r}$, is thus not possible. The possibility of additional constraints, for achieving a determined system of equations, was presented in the literature overview in Section 1.3 but is not investigated in this work. To make full use of the redundancy, the general solution of the underdetermined system, as introduced in (2.64), must be considered. The easiest method to find a solution is to only use the first term of (2.64) by computing the Moore-Penrose pseudo-inverse $\underline{\boldsymbol{G}}^+_\mathrm{a}$, which is the right inverse for the full row rank allocation matrix $\underline{\boldsymbol{G}}_\mathrm{a}$,

$$\underline{\boldsymbol{G}}_\mathrm{a}\underline{\boldsymbol{G}}^+_\mathrm{a} = \underline{\boldsymbol{I}} , \tag{4.42}$$

with the identity matrix $\underline{\boldsymbol{I}}$. The usage of this pseudo-inverse yields the least-squares solution

$${}^\mathrm{s}\boldsymbol{F}^*_{\mathrm{c,T,LS}} = \underline{\boldsymbol{G}}^+_\mathrm{a}\tilde{\boldsymbol{u}}^*_\mathrm{r} . \tag{4.43}$$

It is easy to see that this only represents one of the infinitely many solutions that are described by (2.64). For the tangential forces, the minimum sum of squares is not generally optimal for traction, since the goal is to avoid scenarios such as a soil failure at individual wheels while others don't use their full traction potential. Before introducing the objective function for optimal traction, the allocation equation is derived, which allows to exploit the full solution space for the tangential forces. Therefore, the homogeneous solution of (2.63) is considered. It uses a base of the null-space $\underline{\boldsymbol{G}}^\perp_\mathrm{a}$, which is defined as

$$\underline{\boldsymbol{G}}_\mathrm{a}\underline{\boldsymbol{G}}^\perp_\mathrm{a}\boldsymbol{\rho} = \boldsymbol{0} \quad \forall \quad \boldsymbol{\rho} \in \mathbb{R}. \tag{4.44}$$

Addition of the term $\underline{\boldsymbol{G}}^\perp_\mathrm{a}\boldsymbol{\rho}^*$, with the available DoF trajectory $\boldsymbol{\rho}^*$, to the least-squares solution (4.43) yields the full solution space of (4.40)

$${}^\mathrm{s}\boldsymbol{F}^*_{\mathrm{c,T}} = \boldsymbol{y}^*_\mathrm{w} = \underline{\boldsymbol{G}}^+_\mathrm{a}\tilde{\boldsymbol{u}}^*_\mathrm{r} + \underline{\boldsymbol{G}}^\perp_\mathrm{a}\boldsymbol{\rho}^* . \tag{4.45}$$

By using (4.45) in (4.40) and under consideration of (4.42) and (4.44), it can easily be shown that any choice of the DoF trajectory $\boldsymbol{\rho}^*$ does not change the resulting virtual inputs:

$$\tilde{\boldsymbol{u}}_\mathrm{r} = \underline{\boldsymbol{G}}_\mathrm{a} \left(\underline{\boldsymbol{G}}_\mathrm{a}^+ \tilde{\boldsymbol{u}}_\mathrm{r}^* + \underline{\boldsymbol{G}}_\mathrm{a}^\perp \boldsymbol{\rho}^* \right) = \underbrace{\underline{\boldsymbol{G}}_\mathrm{a} \underline{\boldsymbol{G}}_\mathrm{a}^+}_{\underline{\boldsymbol{I}}} \tilde{\boldsymbol{u}}_\mathrm{r}^* + \underbrace{\underline{\boldsymbol{G}}_\mathrm{a} \underline{\boldsymbol{G}}_\mathrm{a}^\perp \boldsymbol{\rho}^*}_{=\boldsymbol{0}} = \tilde{\boldsymbol{u}}_\mathrm{r}^* \,. \tag{4.46}$$

In this work, the degrees of freedom are adjusted by an optimization algorithm with the objective to minimize the saturation of the traction potential of the contact points. As has been said in Section 3.5, contact force models have high uncertainty, however, their dependency on the normal force is uncontroversial. To be more precise, there is a maximum shear stress at which the soil fails and therefore cannot support any more traction. Mohr-Coulomb (3.33) stated a linear dependency of that maximum shear stress on the normal stress. Therefore, similar to the work of Iagnemma and Dubowsky [ID04a] and Lamon and Siegwart [LS05], the sum of the squared fraction of tangential and normal contact forces is minimized here, yielding the nonlinear optimization problem

$$\min_{\boldsymbol{\rho}^*} \sum_{i=1}^{6} \left(\frac{\| {}^\mathrm{s}\boldsymbol{F}_{\mathrm{c}_i,\mathrm{T}}^*(\boldsymbol{\rho}^*) \|}{F_{\mathrm{c}_i,\mathrm{N}}} \right)^2 . \tag{4.47}$$

This objective function only takes into account changing traction potential that results from normal force shifts between the contact points due to changing orientation and bogie angle configuration. It does not have knowledge about the actual soil states at the individual contact points. Major problems of classical semi-empirical terramechanics models were summarized in Section 3.5.2 and a novel model was developed in Section 3.6. However, problems with its usage in a controller synthesis were shown in Section 3.6.4 and its applicability in the synthesis or the optimization is subject to further research, see the outlook in Section 6.2.

Due to its nonlinear objective function, the optimization (4.47) is solved with the *fmincon* solver in MATLAB and the interior-point algorithm. As long as no effector states or outputs are used in the optimization, a faster optimization algorithm based on the quadratic form of (4.47) could be used. However, it is desired for future work to include wheel quantities to the objective function and/or constraints, which yields more complex nonlinearities and thus requires a fully nonlinear solver. Since solving a nonlinear optimization is computationally costly, the optimization of the DoFs $\boldsymbol{\rho}^*$ is only performed at a much lower update rate compared to the rest of the chassis controller. This is sensible, because the values of $\boldsymbol{\rho}^*$ only change the optimality of the solution but not its correctness when it comes to achieving the virtual input trajectory $\tilde{\boldsymbol{u}}_\mathrm{r}^*$ due to (4.46). The optimality changes with the normal force estimates, i.e. with the rover orientation and bogie angles, see (3.76). Both of these quantities do not change rapidly due to the slow movement of the rover and the assumption of moderately rough terrain, thus a lower sample rate is reasonable. To prevent steps at the rate of the subsampling, the DoF trajectories are filtered with a PT1 filter.

In conclusion, the applied structuring of the solution space for the wheel tangential forces with the explicit separation of the DoFs results in a unique solution in each step. The

DoFs can be adjusted at a lower sampling rate, since they do not compromise the achieved rover-level forces and torque. Together, this separation yields fast runtime of the controller despite the exploitation of the full solution space of the allocation with a potentially complex optimization.

4.4 Wheel Dynamics Feedback Linearization

For the control of the effector force trajectories (4.45), underlying effector controllers are designed to compute the actual drive and steering torque that can be commanded to the motor controllers. These effector controllers are based on the nonlinear combined wheel-ground dynamics model (3.88) which was developed in Section 3.8. One feedback linearization controller is designed for each of the six effectors, i.e. drive, steering and ground contact combinations.

Following the procedure for feedback linearization from Section 2.4, the first derivatives of the outputs $\boldsymbol{y}_{\mathrm{w}_i}$ from (3.88), with the contact forces (3.89) and (3.90), must be computed. For comprehensibility, the outputs of the ith wheel-ground dynamics are repeated here:

$$\boldsymbol{y}_{\mathrm{w}_i} = {}^{\mathrm{s}_i}\underline{\boldsymbol{R}}_{\mathrm{c}_i}(\delta_i) \begin{bmatrix} F_{\mathrm{c}_i,\mathrm{N}} \tan(\phi_{\text{S-S}}) \tanh\left(k_{vx}\left(\omega_i r - \boldsymbol{e}_x^{\top\,\mathrm{c}_i}\underline{\boldsymbol{R}}_{\mathrm{s}_i}(\delta_i)\boldsymbol{v}_{\mathrm{c}_i}(\boldsymbol{x}_{\mathrm{r}}, \boldsymbol{q}_{\mathrm{b}}, \dot{\boldsymbol{q}}_{\mathrm{b}})\right)\right) \\ -F_{\mathrm{c}_i,\mathrm{N}} \tan(\phi_{\text{M-S}}) \tanh\left(k_{vy}\boldsymbol{e}_y^{\top\,\mathrm{c}_i}\underline{\boldsymbol{R}}_{\mathrm{s}_i}(\delta_i)\boldsymbol{v}_{\mathrm{c}_i}(\boldsymbol{x}_{\mathrm{r}}, \boldsymbol{q}_{\mathrm{b}}, \dot{\boldsymbol{q}}_{\mathrm{b}})\right) \end{bmatrix}. \tag{4.48}$$

Due to both outputs being dependent on the states ω_i and δ_i but not $\dot{\delta}_i$, the drive torque input $T_{\omega,i}$ shows up in the first derivative of both outputs, while the steering torque input $T_{\delta,i}$ does not. Both outputs would thus have a relative degree of one, however, since both derivatives only depend on the first input $T_{\omega,i}$, the $\underline{\boldsymbol{V}}$-matrix from (2.35) shows a zero column. According to Lemma 1 on Page 22, this non-regularity yields no relative degree and a feedback linearization with the inversion of the $\underline{\boldsymbol{V}}$-matrix as in (2.44) is not possible.

To circumvent the problem of both outputs' first derivatives being independent of the steering torque $T_{\delta,i}$, the relative degree can be increased. This is done by introducing the drive torque input $T_{\omega,i}$, i.e. the input that does show up in the outputs' first derivatives, as an additional state. Its derivative is used as an input, which leads to the transformed input vector

$$\boldsymbol{u}_{\tilde{\mathrm{w}}_i} = \begin{bmatrix} \dot{T}_{\omega,i} \\ T_{\delta,i} \end{bmatrix} = \begin{bmatrix} \dot{u}_{\mathrm{w}_i,1} \\ u_{\mathrm{w}_i,2} \end{bmatrix}. \tag{4.49}$$

In fact, this adds an integrator to the system dynamics. Details about this technique, which is known as *dynamic extension*, can be found in [Isi95, p. 249ff]. The modified state and input vectors $\boldsymbol{x}_{\tilde{\mathrm{w}}_i}$ and $\boldsymbol{u}_{\tilde{\mathrm{w}}_i}$ are introduced, yielding the extended wheel-ground model of the

ith wheel-ground dynamics

$$\underbrace{\begin{bmatrix}\dot{\omega}_i\\ \dot{\delta}_i\\ \ddot{\delta}_i\\ \dot{T}_{\omega,i}\end{bmatrix}}_{\dot{\boldsymbol{x}}_{\tilde{\mathrm{W}}_i}} = \underbrace{\begin{bmatrix}-\frac{r}{I_{\mathrm{w}_i,y}}F_{\mathrm{c}_i,x}+\frac{1}{I_{\mathrm{w}_i,y}}T_{\omega,i}\\ \dot{\delta}_i\\ 0\\ 0\end{bmatrix}}_{\boldsymbol{f}_{\tilde{\mathrm{W}}_i}} + \underbrace{\begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 1/I_{\mathrm{w}1,z}\\ 1 & 0\end{bmatrix}}_{\underline{\boldsymbol{B}}_{\tilde{\mathrm{W}}_i}} \underbrace{\begin{bmatrix}\dot{T}_{\omega,i}\\ T_{\delta,i}\end{bmatrix}}_{\boldsymbol{u}_{\tilde{\mathrm{W}}_i}} \tag{4.50}$$

with the unchanged output from (4.48). This output is still dependent on ω_i and δ_i but the derivative of ω_i is not a function of an input anymore, since the drive torque T_{ω_i} is a state now. Therefore, the second derivative of the output is needed, in which both elements of the new input vector $\boldsymbol{u}_{\tilde{\mathrm{W}}_i}$ appear. The associated $\underline{\boldsymbol{V}}$-matrix has no zero column anymore and it is regular, hence, a feedback linearization can be performed on the modified wheel-ground dynamics (4.48) and (4.50).

One more peculiarity of the system must, however, be considered before the control law can be stated. Looking at the ith force pair in the output (4.48), it can be seen that the forces depend on the normal force F_{N_i} and the contact point velocity $\boldsymbol{v}_{\mathrm{c}_i}$ additionally to the wheel-ground states and constants. Both quantities can be calculated or estimated through the states of the rigid body dynamics and the measurable bogie angles as indicated in (4.48). For a static contact force model, the whole dynamic could be added to the rigid body dynamics for the synthesis of one dynamic controller, similar to the work of Hoedt [Hoe13]. However, this leads to a rather large system with all effector models and the rigid body dynamics model in one system, whereas the wheel-ground contact must be described as an algebraic constraint, leading to a differential-algebraic system. In this work it is preferred to keep the modular structure that is proposed by Johansen and Fossen [JF13] for easier modification of the individual subsystems, in particular the addition of a dynamic ground contact model in the future. An investigation of the normal forces has shown that they only vary slowly and in a small relative range for the operation of the rover. They are thus assumed piecewise constant for the effector controller. The contact point velocities, on the other hand, show quicker dynamics and must therefore be considered in the synthesis of this controller. Instead of simply applying the standard feedback linearization equations from Section 2.4, the time derivatives are deduced here under consideration of the time-varying velocity $\boldsymbol{v}_{\mathrm{c}_i}$. The first derivative of the output is obtained as

$$\begin{aligned}\dot{\boldsymbol{y}}_{\tilde{\mathrm{W}}_i} &= \frac{\partial \boldsymbol{h}_{\tilde{\mathrm{W}}_i}}{\partial \boldsymbol{x}_{\tilde{\mathrm{W}}_i}}\dot{\boldsymbol{x}}_{\tilde{\mathrm{W}}_i} + \frac{\partial \boldsymbol{h}_{\tilde{\mathrm{W}}_i}}{\partial \boldsymbol{v}_{\mathrm{c}_i}}\dot{\boldsymbol{v}}_{\mathrm{c}_i}\\ &= L_{\boldsymbol{f}_{\tilde{\mathrm{W}}_i}}\boldsymbol{h}_{\tilde{\mathrm{W}}_i} + \underbrace{L_{\underline{\boldsymbol{B}}_{\tilde{\mathrm{W}}_i}}\boldsymbol{h}_{\tilde{\mathrm{W}}_i}}_{=\underline{\mathbf{0}}}\boldsymbol{u}_{\tilde{\mathrm{W}}_i} + \frac{\partial \boldsymbol{h}_{\tilde{\mathrm{W}}_i}}{\partial \boldsymbol{v}_{\mathrm{c}_i}}\dot{\boldsymbol{v}}_{\mathrm{c}_i}\ .\end{aligned} \tag{4.51}$$

As indicated in (4.51), the Lie derivative of the output function along the input matrix is zero and the equation is not a function of the inputs. Therefore, the second derivative is computed as

$$\ddot{\boldsymbol{y}}_{\tilde{\mathrm{W}}_i} = \frac{\partial \dot{\boldsymbol{y}}_{\tilde{\mathrm{W}}_i}}{\partial \boldsymbol{x}_{\tilde{\mathrm{W}}_i}}\left(\boldsymbol{f}_{\tilde{\mathrm{W}}_i} + \underline{\boldsymbol{B}}_{\tilde{\mathrm{W}}_i}\boldsymbol{u}_{\tilde{\mathrm{W}}_i}\right) + \frac{\partial \dot{\boldsymbol{y}}_{\tilde{\mathrm{W}}_i}}{\partial \boldsymbol{v}_{\mathrm{c}_i}}\dot{\boldsymbol{v}}_{\mathrm{c}_i} + \frac{\partial \dot{\boldsymbol{y}}_{\tilde{\mathrm{W}}_i}}{\partial \dot{\boldsymbol{v}}_{\mathrm{c}_i}}\ddot{\boldsymbol{v}}_{\mathrm{c}_i}\ . \tag{4.52}$$

Due to a dependency of the first output derivative (4.51) on the steering angle derivative $\dot{\delta}_i$ and the new drive torque state $T_{\omega,i}$, the second output derivative (4.52) shows a dependency on both inputs $\boldsymbol{u}_{\tilde{\mathrm{w}}_i}$. To apply the feedback linearization control law from Section 2.4, the $\underline{\boldsymbol{V}}$-matrix and $\boldsymbol{w}$-vector are needed. In fact, (2.41) shows that the needed quantities can be obtained from the p_ith derivatives of all outputs where p_i is the relative degree of the ith output and $i = 1 \dots n_y$. In this case, both outputs have a relative degree of two, hence the second derivative (4.52) is used for the deduction. The $\boldsymbol{w}$-vector is the non-input-dependent summand of this second output derivative and thus results from setting $\boldsymbol{u}_{\tilde{\mathrm{w}}_i} = \boldsymbol{0}$ in (4.52). Due to the input-affine form of the system, the input-dependency in the second output derivative is linear, thus the $\underline{\boldsymbol{V}}$-matrix is obtained as the partial derivative of (4.52) with respect to $\boldsymbol{u}_{\tilde{\mathrm{w}}_i}$. Concluding these thoughts, the second output derivative (4.52) can be rewritten as

$$\ddot{\boldsymbol{y}}_{\tilde{\mathrm{w}}_i} = \underbrace{\ddot{\boldsymbol{y}}_{\tilde{\mathrm{w}}_i}(\boldsymbol{u}_{\tilde{\mathrm{w}}_i} = \boldsymbol{0})}_{\boldsymbol{w}_{\tilde{\mathrm{w}}_i}(\boldsymbol{x}_{\tilde{\mathrm{w}}_i}, \boldsymbol{v}_{\mathrm{c}_i}, \dot{\boldsymbol{v}}_{\mathrm{c}_i}, \ddot{\boldsymbol{v}}_{\mathrm{c}_i}, F_{\mathrm{N}_i})} + \underbrace{\frac{\partial \ddot{\boldsymbol{y}}_{\tilde{\mathrm{w}}_i}(\boldsymbol{u}_{\tilde{\mathrm{w}}_i})}{\partial \boldsymbol{u}_{\tilde{\mathrm{w}}_i}}}_{\underline{\boldsymbol{V}}_{\tilde{\mathrm{w}}_i}(\boldsymbol{x}_{\tilde{\mathrm{w}}_i}, F_{\mathrm{N}_i})} \boldsymbol{u}_{\tilde{\mathrm{w}}_i} \tag{4.53}$$

to define the desired matrix $\underline{\boldsymbol{V}}_{\tilde{\mathrm{w}}_i}$ and vector $\boldsymbol{w}_{\tilde{\mathrm{w}}_i}$. A symbolic analysis of $\underline{\boldsymbol{V}}_{\tilde{\mathrm{w}}_i}$, with MATLAB's Symbolic Toolbox, has shown that it is regular, hence a vector relative degree of

$$\boldsymbol{p}_{\tilde{\mathrm{w}}_i} = \begin{bmatrix} 2 & 2 \end{bmatrix} \tag{4.54}$$

is concluded according to Lemma 1. With the $\boldsymbol{w}_{\tilde{\mathrm{w}}_i}$-vector and $\underline{\boldsymbol{V}}_{\tilde{\mathrm{w}}_i}$-matrix, the control law for asymptotic trajectory tracking results, according to the standard procedure from Section 2.4, as

$$\boldsymbol{u}^*_{\tilde{\mathrm{w}}_i} = \underline{\boldsymbol{V}}^{-1}_{\tilde{\mathrm{w}}_i} \left(\ddot{\boldsymbol{y}}^*_{\tilde{\mathrm{w}}_i} - \boldsymbol{w}_{\tilde{\mathrm{w}}_i} - \underline{\boldsymbol{k}}_{\tilde{\mathrm{w}}_i,1} \left(\dot{\boldsymbol{y}}_{\tilde{\mathrm{w}}_i} - \dot{\boldsymbol{y}}^*_{\tilde{\mathrm{w}}_i} \right) - \underline{\boldsymbol{k}}_{\tilde{\mathrm{w}}_i,0} \left(\boldsymbol{y}_{\tilde{\mathrm{w}}_i} - \boldsymbol{y}^*_{\tilde{\mathrm{w}}_i} \right) \right). \tag{4.55}$$

Therein, the controller gain matrices

$$\begin{aligned} \underline{\boldsymbol{k}}_{\tilde{\mathrm{w}}_i,0} &= \mathrm{diag}\left(\begin{bmatrix} k_{\tilde{\mathrm{w}}_i,0,1} & k_{\tilde{\mathrm{w}}_i,0,2} & k_{\tilde{\mathrm{w}}_i,0,3} \end{bmatrix} \right), \\ \underline{\boldsymbol{k}}_{\tilde{\mathrm{w}}_i,1} &= \mathrm{diag}\left(\begin{bmatrix} k_{\tilde{\mathrm{w}}_i,1,1} & k_{\tilde{\mathrm{w}}_i,1,2} & k_{\tilde{\mathrm{w}}_i,1,3} \end{bmatrix} \right) \end{aligned} \tag{4.56}$$

are defined for the feedback of the wheel force errors and their derivatives, respectively. To compute the errors, the trajectories of the effector output $\boldsymbol{y}^*_{\tilde{\mathrm{w}}_i}$ as well as its first and second derivatives $\dot{\boldsymbol{y}}^*_{\tilde{\mathrm{w}}_i}$ and $\ddot{\boldsymbol{y}}^*_{\tilde{\mathrm{w}}_i}$ are needed. Their computation is addressed in Section 4.5.

To ultimately get to the actual actuator trajectories, the input transformation (4.49) is reverted as

$$\boldsymbol{u}^*_{\mathrm{w}_i} = \begin{bmatrix} \int u^*_{\tilde{\mathrm{w}}_i,1} \mathrm{d}t \\ u^*_{\tilde{\mathrm{w}}_i,2} \end{bmatrix}. \tag{4.57}$$

No internal dynamics result for the closed-loop wheel-ground system, since the sum of the vector relative degree elements from (4.54) equals the system order $n_{\tilde{\mathrm{w}}_i}$:

$$\sum_{j=1}^{2} p_{\tilde{\mathrm{w}}_i,j} = 4 = n_{\tilde{\mathrm{w}}_i}\,. \tag{4.58}$$

4.5 Trajectory Derivatives

For the feedback law of the effector controller (4.55), the first two derivatives of the output trajectories $\dot{\boldsymbol{y}}^*_{\tilde{\mathrm{w}}_i}$ and $\ddot{\boldsymbol{y}}^*_{\tilde{\mathrm{w}}_i}$ are needed. In Section 4.3, however, only the wheel force trajectories $\boldsymbol{y}^*_{\tilde{\mathrm{w}}_i}$ themselves were computed with (4.45). Its first and second time derivatives are

$$\dot{\boldsymbol{y}}^*_{\tilde{\mathrm{w}}_i} = \frac{\mathrm{d}\underline{\boldsymbol{G}}^+_{\mathrm{a}}}{\mathrm{d}t}\tilde{\boldsymbol{u}}^*_{\mathrm{r}} + \underline{\boldsymbol{G}}^+_{\mathrm{a}}\dot{\tilde{\boldsymbol{u}}}^*_{\mathrm{r}} + \frac{\mathrm{d}\underline{\boldsymbol{G}}^\perp_{\mathrm{a}}}{\mathrm{d}t}\boldsymbol{\rho}^* + \underline{\boldsymbol{G}}^\perp_{\mathrm{a}}\dot{\boldsymbol{\rho}}^* \,, \tag{4.59}$$

$$\ddot{\boldsymbol{y}}^*_{\tilde{\mathrm{w}}_i} = \frac{\mathrm{d}^2\underline{\boldsymbol{G}}^+_{\mathrm{a}}}{\mathrm{d}t^2}\tilde{\boldsymbol{u}}^*_{\mathrm{r}} + 2\frac{\mathrm{d}\underline{\boldsymbol{G}}^+_{\mathrm{a}}}{\mathrm{d}t}\dot{\tilde{\boldsymbol{u}}}^*_{\mathrm{r}} + \underline{\boldsymbol{G}}^+_{\mathrm{a}}\ddot{\tilde{\boldsymbol{u}}}^*_{\mathrm{r}} + \frac{\mathrm{d}^2\underline{\boldsymbol{G}}^\perp_{\mathrm{a}}}{\mathrm{d}t^2}\boldsymbol{\rho}^* + 2\frac{\mathrm{d}\underline{\boldsymbol{G}}^\perp_{\mathrm{a}}}{\mathrm{d}t}\dot{\boldsymbol{\rho}}^* + \underline{\boldsymbol{G}}^\perp_{\mathrm{a}}\ddot{\boldsymbol{\rho}}^* \,. \tag{4.60}$$

Due to their complexity, the derivatives of the pseudo-inverse and especially the null-space base $\underline{\boldsymbol{G}}^\perp_{\mathrm{a}}$ are very computationally expensive and badly conditioned. Therefore the derivatives of $\boldsymbol{y}^*_{\tilde{\mathrm{w}}_i}$ are computed by filtering instead of with (4.59) and (4.60).

4.6 Controller Overview

The full chassis controller was derived as three sub-controllers in this chapter. For a better overview, the full controller layout is depicted in Figure 4.4, where the three sub-controllers are marked with bold gray boxes.

Starting at the very left, the input to the chassis controller is the rover velocity trajectory $\boldsymbol{y}^*_{\mathrm{r}}$ and its derivative $\dot{\boldsymbol{y}}^*_{\mathrm{r}}$. Additionally, the measured rover states $\boldsymbol{x}_{\mathrm{r}}$, bogie angles $\boldsymbol{q}_{\mathrm{b}}$ and bogie rates $\dot{\boldsymbol{q}}_{\mathrm{b}}$ are used in multiple blocks. The corresponding control law is derived in (4.22). Following the top signal line of $\dot{\boldsymbol{y}}^*_{\mathrm{r}}$, three terms are added to it: At the first summation, the rover velocity error, multiplied by the gain matrix $\underline{\boldsymbol{k}}_{\mathrm{r},0}$, is added. The second summation adds the integral of this error, multiplied by another gain matrix $\underline{\boldsymbol{k}}_{\mathrm{r,I}}$, and finally, the linearization term $-\underline{\boldsymbol{C}}_{\mathrm{r}}\tilde{\boldsymbol{f}}_{\mathrm{r}}$ is added. The latter consists of the output matrix $\underline{\boldsymbol{C}}_{\mathrm{r}}$, the $\boldsymbol{f}_{\mathrm{r}}$-function of the rigid body dynamics and the multiplication of the normal forces with parts of the Jacobian and the input matrix, see (4.14). For better readability, the optional anti-windup for the integrator from Section 4.2.1 is not shown in the diagram. Below the rover level controller box, the estimation (3.76) of the normal forces $\boldsymbol{F}_{\mathrm{c,N}}$ from RPY and bogie angles as well as the contact point velocity computation (3.5) from states, bogie angles and bogie rates is shown. To obtain the virtual input trajectory $\tilde{\boldsymbol{u}}^*_{\mathrm{r}}$, i.e. the rover level forces $F_{\mathrm{r},x}$, $F_{\mathrm{r},y}$ and $T_{\mathrm{r},z}$, the whole term is multiplied with the inverted $\underline{\boldsymbol{V}}_{\mathrm{r}}$-matrix.

The allocation consists of the summation of least-squares solution $\underline{\boldsymbol{G}}^+_{\mathrm{a}}\tilde{\boldsymbol{u}}^*_{\mathrm{r}}$ and the homogeneous solution $\underline{\boldsymbol{G}}^\perp_{\mathrm{a}}\boldsymbol{\rho}^*$ with the DoF trajectory $\boldsymbol{\rho}^*$. The latter is obtained through the optimization (4.47), which is only performed at a lower update rate and the old DoFs are used in the meantime, see Section 4.3. Through this sum, the wheel tangential contact force trajectories $\boldsymbol{y}^*_{\mathrm{w}}$ are computed in the non-steered coordinate systems and subsequently fed to the underlying wheel-ground controller.

One wheel-ground controller exists for each of the six wheels, which is indicated by the multi-layered gray box and can be seen at the index $i = 1 \ldots 6$ that is used throughout

this sub-controller. The force trajectory $\boldsymbol{y}^*_{\tilde{\mathrm{w}}_i}$ is filtered and derived with respect to time twice to obtain $\dot{\boldsymbol{y}}^*_{\tilde{\mathrm{w}}_i}$ and $\ddot{\boldsymbol{y}}^*_{\tilde{\mathrm{w}}_i}$ as described in Section 4.5. Following (4.55), the second force derivative trajectory is then enhanced with three summations, similar to the rover level controller. First, the force derivative error $(\dot{\boldsymbol{y}}^*_{\tilde{\mathrm{w}}_i} - \dot{\boldsymbol{y}}_{\tilde{\mathrm{w}}_i})$ is multiplied with the gain matrix $\underline{\boldsymbol{k}}_{\tilde{\mathrm{w}}_i,1}$ and added. For the computation of the actual force derivatives, (4.51) is used which consists of the shown Lie and partial derivatives. In the second summation, the error in the forces themselves is multiplied with the gain matrix $\underline{\boldsymbol{k}}_{\tilde{\mathrm{w}}_i,0}$ and finally the linearization term $\boldsymbol{w}_{\tilde{\mathrm{w}}_i}$ is subtracted. For the computation of these three summands, the first and second time derivatives of the contact point velocity $\boldsymbol{v}_{\mathrm{c}_i}$ are needed, which are computed numerically after filtering the velocity to ensure smoothness of the derivatives. The modified wheel input trajectory $\boldsymbol{u}_{\tilde{\mathrm{w}}_i}$ is obtained by a multiplication with the normal force and wheel state dependent, inverted $\underline{\boldsymbol{V}}_{\tilde{\mathrm{w}}_i}$-matrix and the reverse input transformation (4.57) is finally used to get the drive and steering torque trajectory $\boldsymbol{u}^*_{\mathrm{w}}$.

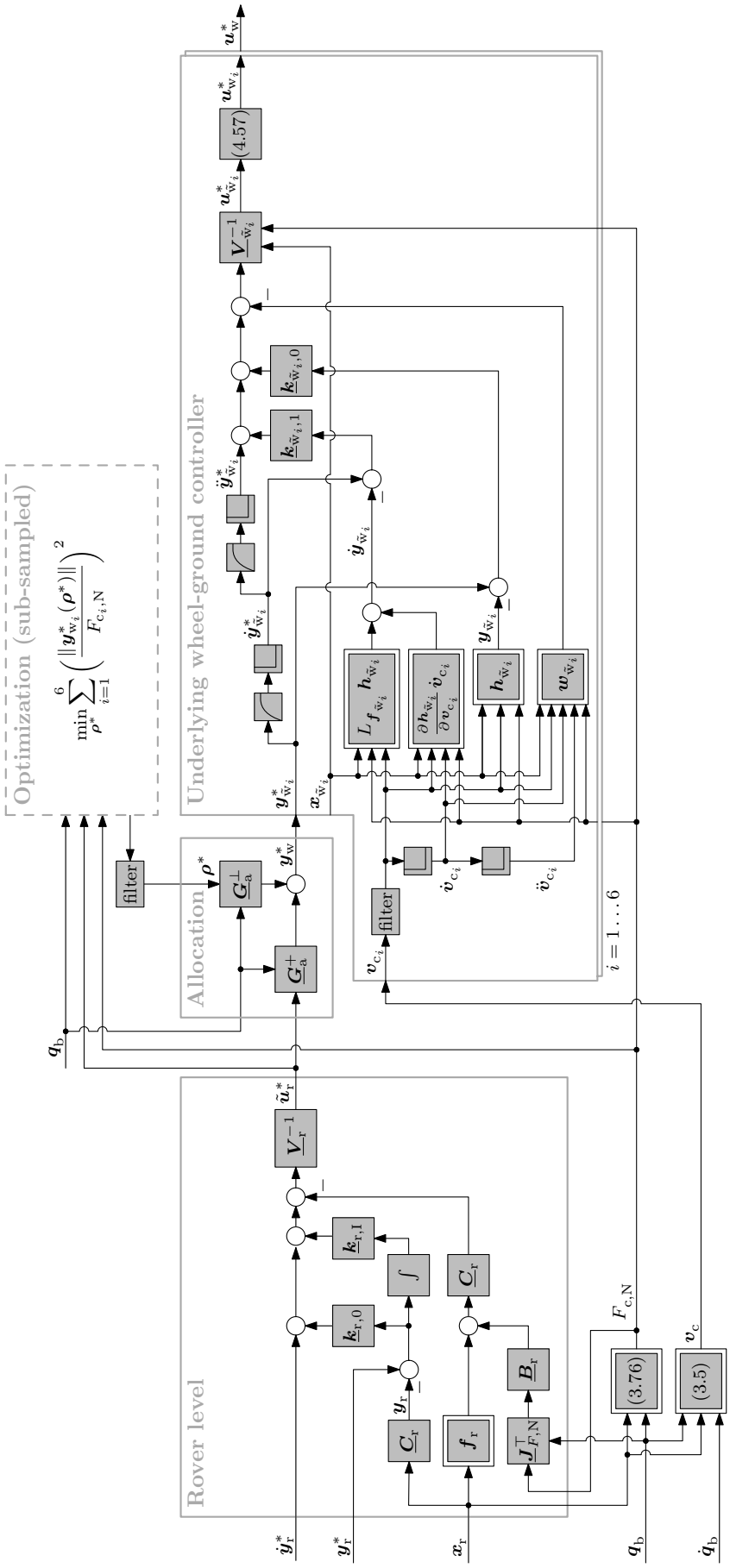


Figure 4.4: Full chassis controller overview.

5 Simulative Verification

A simulative verification of the chassis control system, that was derived in Chapter 4, is presented in this chapter. Extensive work has been done at our institute in the area of multi-physics modeling of planetary exploration rovers and their wheel-ground contact. Section 5.1 gives a brief overview of this work and the ExoMars rover and terramechanics model that is used in this work in particular. The verification itself comprises a nominal case without major disturbances in Section 5.2. A test of the controller robustness to contact model errors and disturbances, in the form of non-zero contact angles, is shown in Section 5.3. This chapter is concluded with a discussion of the strengths and weaknesses of the controller in Section 5.4.

5.1 High-Fidelity Simulation Model and Framework

In the last five years, the *DLR Rover Simulation Toolkit (RST)* [HBB17] was developed at our institute and used in a number of projects with very different purpose and level of detail. To enable modeling of the different physical domains with high flexibility, the RST is implemented in the multi-physics modeling language *Modelica*. The latter provides multi-body mechanics, electrical and even thermal and fluid components in its standard library [EMO99]. The RST builds upon these standard components as well as the in-house *Contact Dynamics Library* for different tier models for interaction between wheels or other objects and rigid surfaces or various sand types [Lic+16].

Base models combine components from the standard library in a modular way to create general building blocks for the modeling of rovers or other mobile robots. Due to the parallel kinematic chains with passive and active rotational degrees of freedom, the main base model is what we called the *fork*. It consists of one rotational degree of freedom with one mechanical flanch on the input and one (*1-fork*) or two (*2-fork*) flanches on the output side, where all of the mass and dimension properties are configurable. Interchangeable gear, motor, servo and sensor models, including optional rotational flexibilities in the drive train, allow to choose from different fidelity models for each of the listed components. This enables simple and individual adaption to various use cases. Available motor models, to give an example, range from ideal torque or position sources to motor models with detailed electrical, mechanical and thermal sub-models, provided by the Modelica standard library. One special feature is that each fork has a bus connector for communication with the other components and the control software or a co-simulation interface. This bus is pre-configured with all sensor signals and command values and connects all entities of forks, additional sensors and main controller. To simulate a realistic communication, the bus is sampled with the

controller frequency and it provides a well-defined interface to interact with controllers in other modeling languages such as MATLAB/Simulink.

Implemented rover models in the RST include the ExoMars, JPL's Curiosity, the DLR Leightweight Rover Unit [Wed+15a], the DLR Scout Rover [Wie18] and the MMX Rover [Sed+20]. Applications range from high-level interactive mission demonstration [Wed+15b] over the Software-in-the-loop development of autonomous navigation software with images from our 3D visualization [HBB17] to in-depth traction performance evaluation of Curiosity in a project with NASA/JPL [Bus+18]. Recently, the MMX rover design started and the RST is used in all project domains that need system simulation including systems engineering, locomotion subsystem development and operations [Sed+20]. Details about the implementation and selected applications can be found in [HBB17].

The proposed controller is implemented in MATLAB/Simulink, where all functions were created with the MATLAB Symbolic Toolbox and subsequently converted to standard MATLAB functions for better performance. A User Datagram Protocol (UDP) communication with a rate of 1 kHz is established between Simulink and the RST in Modelica to exchange sensor signals and command values. Due to the blocking property of the used UDP communcation blocks, it is ensured that the two simulations wait for one another in each communication step, such that none of them runs ahead of the other.

5.2 Verification and Comparison to a Constraint-based Controller

This first verification section shows the performance of the proposed controller, which is called *All-Terrain Controller 2 (ATC2)* in the remainder, for relatively small model errors and disturbances. The name is chosen following the name All-Terrain Controller from an earlier rover chassis controller [BZ17], although the whole structure and most sub-models have changed completely. To show the benefits of ATC2, the results are compared with a state of the art constraint-based controller that calculates steering angles and wheel rates based on an ICR, see Section 5.2.1. The simulated scenario and the most important model parameters as well as all controller parameters are introduced in Section 5.2.2, before showing the results of the co-simulation in Section 5.2.3.

5.2.1 Reference Controller

For the purpose of comparing the proposed chassis controller to the state of the art, the most widely used constraint-based ICR control is implemented. This control strategy uses the same trajectory input as the proposed ATC2 of this work, i.e. the forward, sideways and yaw velocity. In contrast to ATC2, however, rates for the drive motors and angles for the steering actuators are commanded. The underdeterminacy of the system of equations,

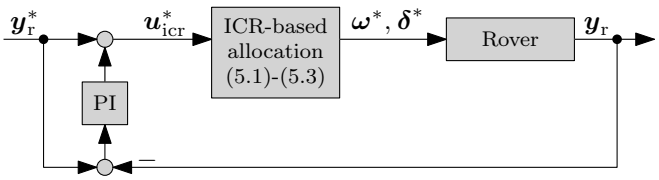


Figure 5.1: Layout of the ICR-based reference controller.

that relates rover velocities to the desired steering angles and wheel rates, is solved by the introduction of additional constraints. These constraints are derived geometrically from a zero slip assumption for the wheels and a required ICR, which gives the controller its name, see Figure 1.5b. This allows explicit computation of the actuator commands from the system-level velocity trajectories with

$$\delta_{i,\mathrm{icr}} = \tan^{-1}(v_{i,y}^*/v_{i,x}^*) \;, \tag{5.1}$$

$$\omega_{i,\mathrm{icr}} = \frac{1}{r}\|\boldsymbol{v}_i^*\|_2 \;, \tag{5.2}$$

where the index $(\cdot)_\mathrm{icr}$ indicates quantities of the reference ICR controller. The desired velocity of the ith wheel $\boldsymbol{v}_i^*$ is calculated from the rover level input trajectory $\boldsymbol{u}_\mathrm{icr}^*$ as

$$\boldsymbol{v}_i^* = \begin{bmatrix} u_{\mathrm{icr},1}^* \\ u_{\mathrm{icr},2}^* \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_{\mathrm{icr},3}^* \end{bmatrix} \times \boldsymbol{r}_{\mathrm{w}_i|\mathrm{r}}(\boldsymbol{q}_\mathrm{b}) \;. \tag{5.3}$$

Although this is a 3-dimensional vector, the third component is always zero, since there is no desired rover velocity in the vertical direction, nor are there desired pitch and roll trajectories. To obtain the rover input trajectory $\boldsymbol{u}_\mathrm{icr}^*$, a PI-controller is added to the feed forward desired rover velocity $\boldsymbol{y}_\mathrm{r}^* = [v_{\mathrm{r},x}^*, v_{\mathrm{r},y}^*, \omega_{\mathrm{r},z}^*]^\top$ to compensate for wheel slip, yielding

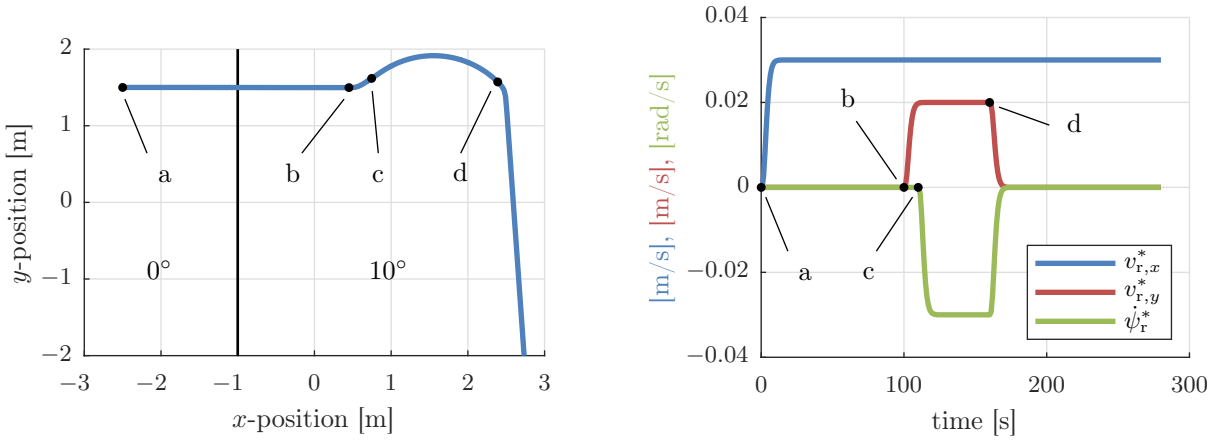
$$\boldsymbol{u}_\mathrm{icr}^* = \boldsymbol{y}_\mathrm{r}^* + \underline{\boldsymbol{k}}_{\mathrm{icr,P}}\,(\boldsymbol{y}_\mathrm{r}^* - \boldsymbol{y}_\mathrm{r}) + \underline{\boldsymbol{k}}_{\mathrm{icr,I}} \int (\boldsymbol{y}_\mathrm{r}^* - \boldsymbol{y}_\mathrm{r})\,\mathrm{d}t \;. \tag{5.4}$$

$\underline{\boldsymbol{k}}_{\mathrm{icr,P}}$ and $\underline{\boldsymbol{k}}_{\mathrm{icr,I}}$ are the proportional and integral gain diagonal matrices, respectively. The layout of the reference controller is depicted in Figure 5.1.

This reference controller enables an explicit computation of wheel rates and steering angles, however, omitting any model knowledge and assuming zero slip and planar movement. All such disturbances must be covered by the PI-controller which will be seen in comparing the results to ATC2 in Section 5.2.3.

5.2.2 Scenario Description

For this first verification, the synthesis model from Section 3.7 is chosen as the ground contact model in the RST, i.e. there are no modeling errors in the contact force model. The rover itself, however, is modeled as a fully dynamic multi-body model, including the



(a) Driven path of the rover with indicated changes in the input trajectory.

(b) Desired rover trajectory in forward, sideways and yaw velocity.

Figure 5.2: Path and rover velocity trajectories for the scenario of Section 5.2.

dynamics of the bogies. Additionally, realistic torque and rate limits, which are not considered in the controller synthesis, are set for the steering and drive actuators. A fixed velocity trajectory is chosen and processed through a fourth order filter to make the trajectory four times continuously differentiable and smooth enough for the slow dynamics of the ExoMars actuators. Especially since the contact force and actuator limits are not considered in the controller, the trajectory must be such that it is feasible within the limits. The raw trajectory filter input consists of steps in the forward, sideways and yaw velocity at 0 s, 100 s, 110 s and 160 s and the filtered output can be seen in Figure 5.2b, where said times are marked with a black dot and labeled with 'a' to 'd'. In terms of the ground, an even surface with a kink from an inclination of 0° to 10° at an x-position of -1 m, see Figure 5.2a, is generated. At the rover start position 'a', the rover accelerates on the non-inclined plane and subsequently hits the kink with its front, middle and rear wheel pairs. Being completely on the 10° slope, the rover then starts to drive sideways from 'b' on and a curve from 'c' on until it has rotated by almost 90° at 'd'. It is then commanded to drive purely forward again. Due to the directions being always expressed in the rover-fixed frame, this results in driving almost orthogonally to the slope gradient.

The kink represents a large disturbance since it leads to contact angles of up to 10°, while zero contact angle was assumed for the controller design. To separate between the controller performance for small model errors and disturbances on one side, and the robustness against large disturbances on the other, the scenario is divided into two parts. A first part (0 s-90 s), which contains the large disturbance of the ground kink, is presented in Section 5.3 and a second part (90 s-200 s), where the full rover is on the 10° inclination already, is shown in Section 5.2.3.

The most important model parameters can be found in Table 5.1 with their used value and a reference to the equation(s) where they were introduced. With the same systematic, the controller parameters are summarized in Table 5.2.

Table 5.1: Model parameters as used for the scenario in Section 5.2.

Model parameter	Value	Used in Equation
Rigid body mass m_r	$100\,\mathrm{kg}$	(3.12)
Rigid body inertia $\underline{\boldsymbol{I}}_\mathrm{r}$	$\mathrm{diag}([37.5, 37.5, 37.5])$	(3.26),(4.16)
Wheel inertia steering axis $I_{\mathrm{w}_i,z}$	$2.28\,\mathrm{kg\,m^2}$	(3.88)
Wheel inertia drive axis $I_{\mathrm{w}_i,y}$	$5.40\,\mathrm{kg\,m^2}$	(3.88)
Friction angle sand-sand $\phi_\mathrm{S\text{-}S}$	$28°$	(3.89)
Friction angle metal-sand $\phi_\mathrm{M\text{-}S}$	$15°$	(3.90)
Longitudinal contact force parameter k_{vx}	10	(3.89)
Lateral contact force parameter k_{vy}	10	(3.90)
Maximium resistance torque $T_\mathrm{res,max}$	1	(3.88)
Steer resistance torque parameter k_δ	10	(3.88)
Wheel radius r	$0.125\,\mathrm{m}$	(3.88),(3.89)

Table 5.2: Controller and reference controller parameters as used for the scenario in Section 5.2.

Controller parameter	Value	Used in Equation
Rigid body dynamics gain $\underline{\boldsymbol{k}}_{\mathrm{r},0}$	$\mathrm{diag}([12, 4, 4])$	(4.23)
Rigid body dynamics gain $\underline{\boldsymbol{k}}_{\mathrm{r,I}}$	$\mathrm{diag}([50, 15, 8])$	(4.23)
Wheel-ground dynamics gain $\underline{\boldsymbol{k}}_{\mathrm{w}_i,0}$	$\mathrm{diag}([1000, 1000])$	(4.55)
Wheel-ground dynamics gain $\underline{\boldsymbol{k}}_{\mathrm{w}_i,1}$	$\mathrm{diag}([24, 24])$	(4.55)
Reference Controller gain $\underline{\boldsymbol{k}}_{\mathrm{icr,P}}$	$\mathrm{diag}\,[0.1, 0.1, 0.1]$	(5.4)
Reference Controller gain $\underline{\boldsymbol{k}}_{\mathrm{icr,I}}$	$\mathrm{diag}\,[2, 2, 2]$	(5.4)

5.2.3 Simulation Results

In this subsection, the results of the rover drive on the 10° inclination are shown, which corresponds to the time period 90 s-200 s of the full scenario from Section 5.2.2. The analysis of the results is structured according to the controller structure, i.e. the rover level rigid body controller performance is shown first, followed by the allocation and the underlying wheel-ground dynamics controller.

Rover level. Figure 5.3 summarizes the rover motion during its maneuver from straight uphill drive to overlayed sideways and yaw velocities, leading to a turn of close to 90°, back to straight driving almost orthogonal to the slope. This corresponds to the positions 'b', 'c' and 'd' in the path plot in Figure 5.2. The bold blue plot is the velocity trajectory $\boldsymbol{y}_\mathrm{r}^*$, the red one represents the measured velocity of the ATC2-controlled rover and the green 'ICR' plot shows the velocity of the rover controlled with the reference ICR controller. The three subplots represent the three components of the rover velocity trajectory, namely the forward velocity $v_{\mathrm{r},x}$, the sideways velocity $v_{\mathrm{r},y}$ and the yaw rate $\omega_{\mathrm{r},z}$. Although small deviations can

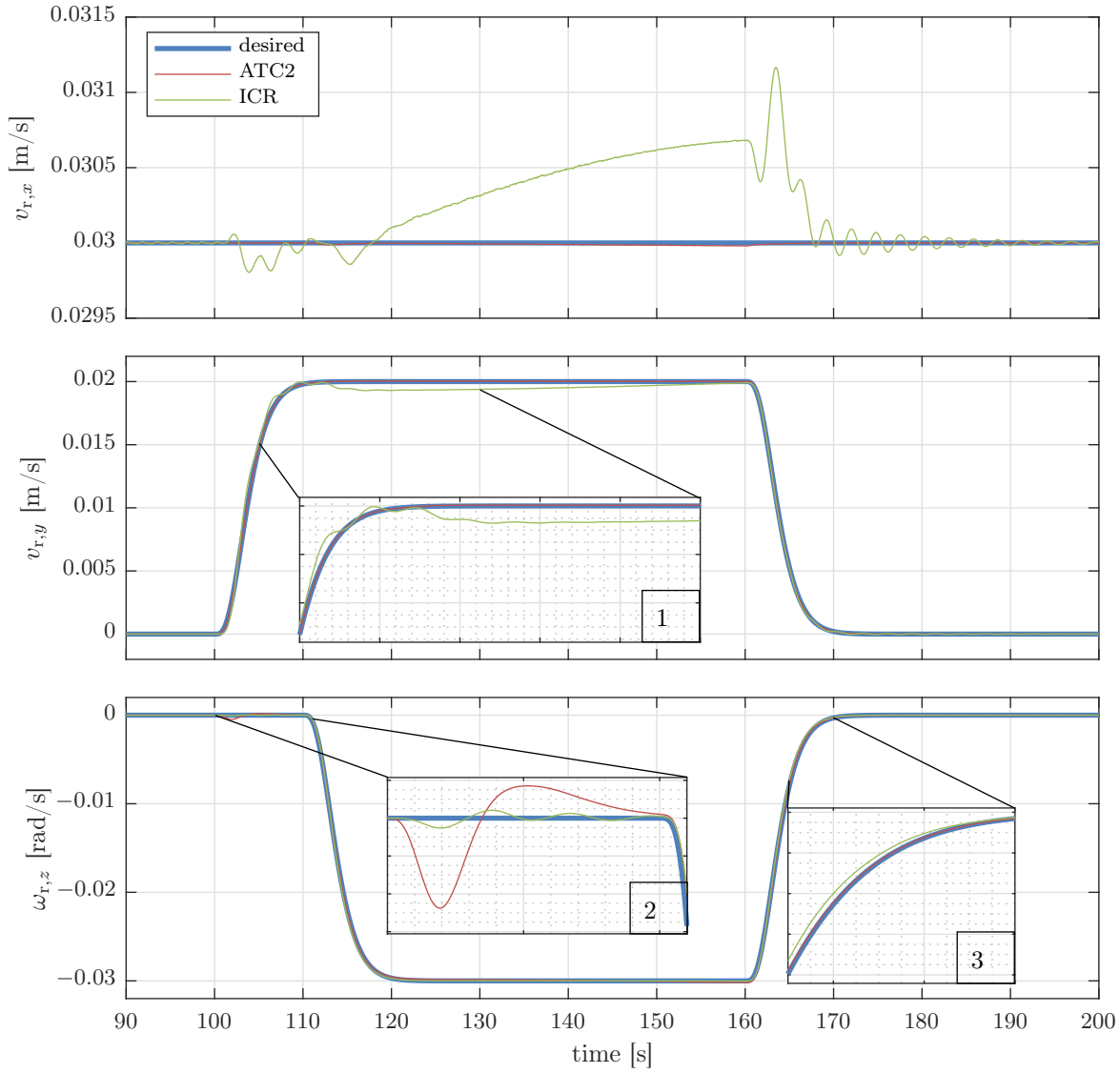
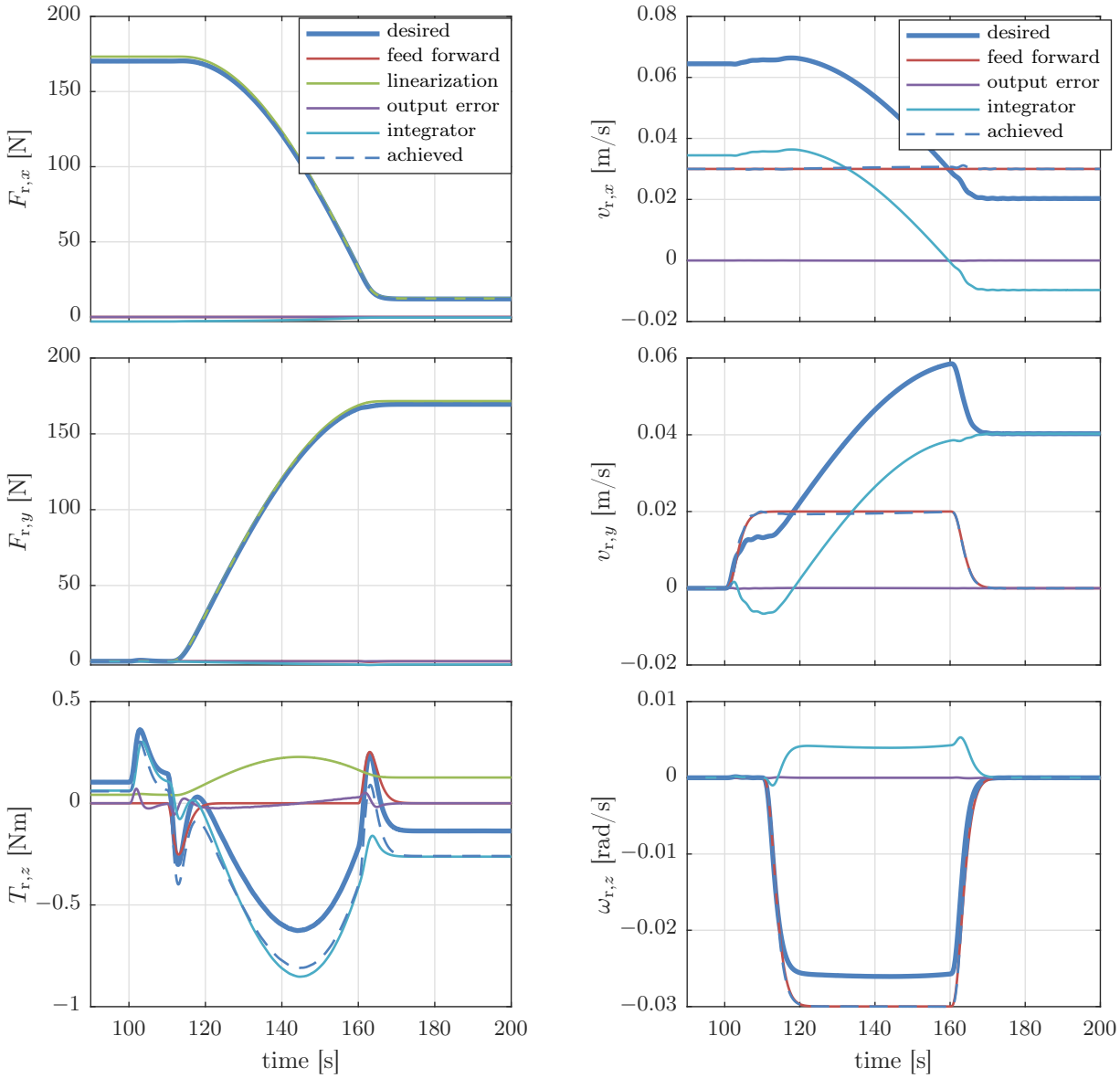


Figure 5.3: Trajectory following of the proposed ATC2 controller compared to the reference ICR controller.

be observed, the overall trajectory following behavior of both controllers is decent. The first subplot is largely zoomed in, note the scale of the y-axis. The ATC2 behavior only shows more deviation than the ICR in the yaw rate at 100 s-110 s, see magnification '2'. ICR, on the other side, shows more deviation in the forward and sideways velocities, see the first two subplots including magnification '1', and slightly worse performance in the yaw velocity during the change of the desired yaw velocity, see magnification '3'. This can be explained with the necessity of a rather aggressive controller setting and large integrator action, details on that are explained in the following by looking deeper into the components of the rover level parts of the controllers.



(a) ATC2 (b) ICR

Figure 5.4: Output of the rover level controllers of ATC2 and ICR with a breakdown into their respective components.

The difference between ATC2 and ICR is investigated in Figure 5.4, where the outputs of the rover level controllers of both approaches are compared. The outputs of the ICR's PI-controller with feed forward is shown in Figure 5.4b and Figure 5.4a depicts the outputs of ATC2's rigid body dynamics feedback linearization. Note that the units of the two are different: Due to the reference ICR controller's velocity-based allocation, the outputs of its rover level controller (5.4) are velocities as well. ATC2, on the other side, needs the force/torque virtual inputs for its force-based allocation as described in Section 4.1. In both figures, the respective total trajectory is plotted as a bold blue line, whereas the thin lines of different colors represent the different components of it, see the legends for details. For an

easier understanding of the ATC2 terms, the control law (4.23) is repeated here and labeled according to the legend of Figure 5.4:

$$\tilde{\boldsymbol{u}}_{\mathrm{r}}^{*} = \underline{\boldsymbol{V}}_{\mathrm{r}}^{-1} \left(\underbrace{-\underline{\boldsymbol{C}}_{\mathrm{r}} \tilde{\boldsymbol{f}}_{\mathrm{r}}}_{\text{linearization}} \quad \underbrace{+\dot{\boldsymbol{y}}_{\mathrm{r}}^{*}}_{\text{feed forward}} \quad \underbrace{-\underline{\boldsymbol{k}}_{\mathrm{r,0}} \left(\boldsymbol{y}_{\mathrm{r}} - \boldsymbol{y}_{\mathrm{r}}^{*} \right)}_{\text{output error}} \underbrace{-\underline{\boldsymbol{k}}_{\mathrm{r,I}} \int \left(\boldsymbol{y}_{\mathrm{r}} - \boldsymbol{y}_{\mathrm{r}}^{*} \right) \mathrm{d}t}_{\text{integrator}} \right) . \tag{5.5}$$

The colors for ICR in Figure 5.4b are analogous, however, there is no model-based linearization term in the reference controller (5.4). It can be seen in Figure 5.4 that high forces in forward and sideways direction are commanded, whereas the yaw torque stays below an absolute value of 1 N m. This stems from the 10° slope that the rover is driving on, which requires large forces to counteract the downhill force but has no influence on the torque that is required to cause a yaw motion. The compensation of the downhill force first causes a large x-force while the rover drives straight uphill and slowly shifts to almost pure y-direction once the rover has finished its turn and drives almost orthogonal to the slope. The y-force then compensates the gravity and prevents the rover from sliding sideways downhill, enabled by the all-wheel steering, which will be seen in the wheel-wise considerations. Looking at the force breakdown, i.e. the thin lines in Figure 5.4, it can be seen that the large x- and y-forces are almost purely covered by the linearization term (green line) for the ATC2 case. This part is the one that compensates the nonlinear system model, including the gravity forces, by measurement of the rover's states and its underlying 3-dimensional model. The required yaw torque $T_{\mathrm{r},z}$ is so small that the relatively little modeling errors can be observed. These errors are mainly the neglected bogie dynamics, inaccuracies in the CoG and inertia of the assumed rigid body as well as the quasi-static normal force estimation, which does not take the tangential contact forces into account. In contrast to the model-based ATC2, the trajectory tracking of the reference ICR controller is largely dependent on the integrator part of the PI-controller, as can be seen in Figure 5.4b. Therein, the cyan plot shows the integrator fraction of the total rover velocity that is fed to the ICR algorithm. The large deviations from the feed forward solution can easily be explained with the missing knowledge of the gravity direction and thereby the downhill force in combination with a zero-slip assumption of the following ICR algorithm. This algorithm simply computes steering angles and wheel rates such that the desired rover motion would be achieved with zero slip, see Section 5.2.1 for more details. The integrator action needs to compensate these errors but generally leads to a slow response to rapidly changing conditions. To still achieve a decent trajectory following behavior, the PI-controller gains need to be tuned rather aggressively, which in turn leads to the oscillations that where observed in Figure 5.3.

The dashed lines in Figure 5.4 show how well the remainder of the controller, i.e. the ICR algorithm for the reference controller and the allocation and underlying wheel controller for ATC2, is able to achieve the desired rover-level forces (ATC2) or velocities (ICR). For ICR, the errors, that are caused by the zero slip assumption, can be seen in terms of a large discrepancy between desired and achieved velocity. In fact, the total desired velocity for the ICR algorithm needs to be increased considerably by the error integrator to achieve the actually desired velocity profile.

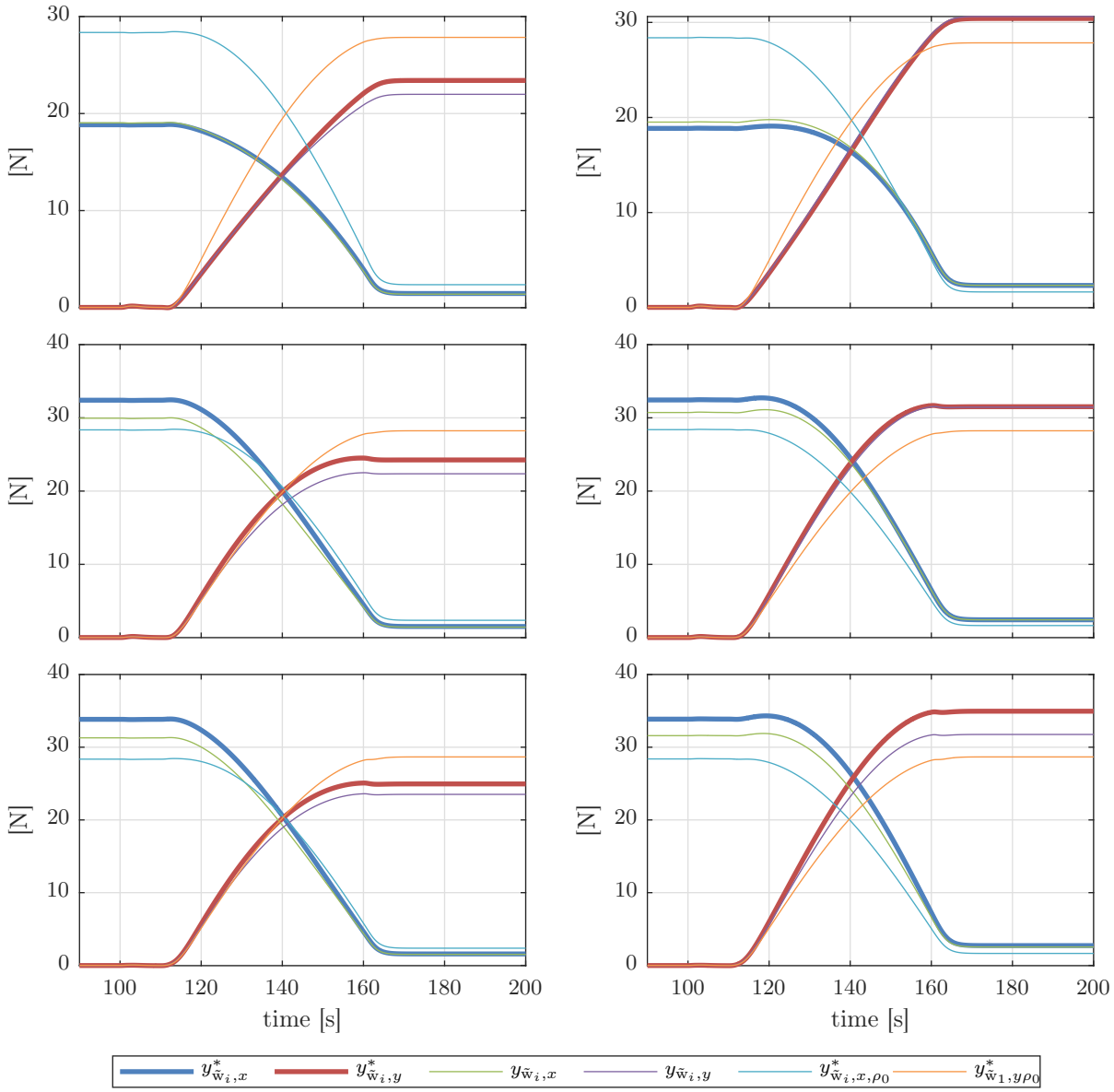


Figure 5.5: Wheel force trajectories in x- and y-direction (bold lines), the trajectories for $\boldsymbol{\rho}^* = \mathbf{0}$ (cyan and orange) and the achieved forces (green and purple).

Allocation. For the assessment of the allocation part of the ATC2 controller, the desired wheel forces for all six wheels are plotted in Figure 5.5. Bold lines are again trajectories, blue represents the x-force and red the y-force. Both are expressed in their respective non-steered 's_i'-frames, which have the same orientation as the main rover frame. This means that x always points in the rover forward direction and y sideways, independent of the individual steering angles. Additionally to these optimized force trajectories, the least-squares solution, i.e. the allocation solution for $\boldsymbol{\rho}^* = \mathbf{0}$, is plotted as thin cyan and orange lines for the x- and y-direction, respectively. The effect of the optimization can be seen very clearly by comparing the x-force trajectory in blue and cyan in the sections 90 s-110 s. While the least-squares solution (thin cyan lines) distributes the forces equally to all six wheels, the optimization

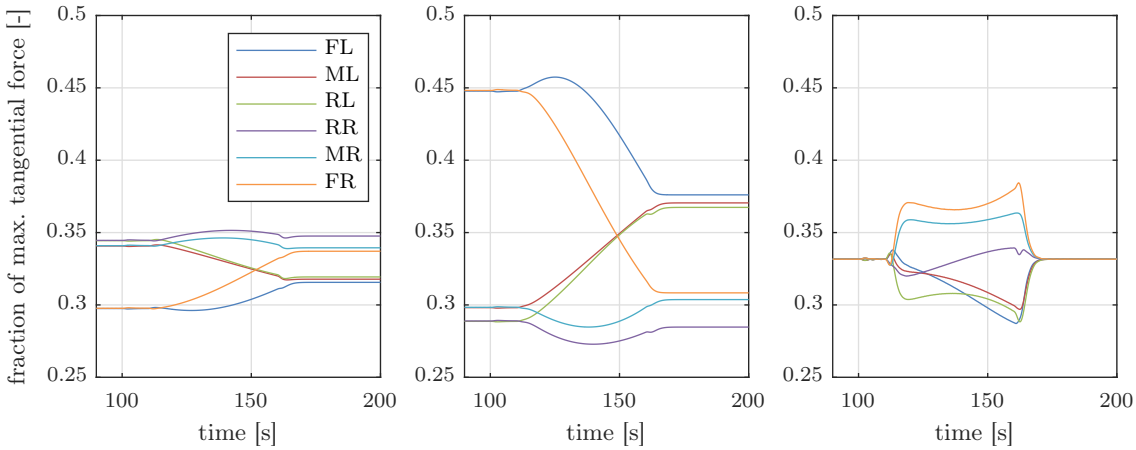


Figure 5.6: Fraction of maximum tangential force that is used per wheel. First letter stands for Front, Middle, Rear and second letter for Left or Right.

shifts more traction forces to the back and middle wheels and lowers the desired force at the front wheels. This corresponds to the estimated normal contact forces that increase at the back wheels and decrease at the front wheels due to the slope. Between 110 s and 160 s, the rover turns by almost 90°, which leads to a roll angle of the rover with its right wheels lower than the left wheels, see Figure 5.13 for a 3D visualization of the rover during the turn. This is again reflected in the normal force estimation and leads to the optimization distributing more traction forces onto the right wheels than the left ones. In Figure 5.6, the fraction of used traction potential

$$\zeta_i = \frac{\|\boldsymbol{F}_{c_i,T}\|_2}{F_{N,i}} \tag{5.6}$$

is plotted for ATC2 (left), ATC2 without optimization, i.e. the least squares solution, (middle) and the reference ICR controller (right). The benefit of optimizing the force allocation compared to using the least squares solution can clearly be seen in a traction potential usage difference of up to 10 %. It is pointed out again, that the higher the margin to the maximum possible traction force per wheel, the lower the chance of soil failure and excessive wheel slip for the largely uncertain soil behavior. The traction potential usage of the reference controller is computed during post processing, since the ICR controller itself does not use and therefore does not compute the traction forces. The same ATC2 synthesis ground contact model is used to compute the actual traction forces for the ICR from the wheel and rover states. Looking at the ICR (right plot in Figure 5.6) it can be seen that during straight drives, i.e. with equal wheels rates, the traction potential usage is identical for all wheels. This results from the used contact model, see Section 3.7.3, which assumes the traction force as the product of maximum traction force and a wheel slip dependent coefficient. For the identical wheel slip of the straight drives with ICR, this results in equal coefficients and therefore equal tangential forces. While this is very convenient without disturbances, it leads to a less optimal share for changing grounds or maneuvers with differing wheel slip as for example the turn at 110 s-160 s. ATC2's allocation, on the other side, is able to keep the saturation of the wheels at similar levels to within 6 % for this scenario.

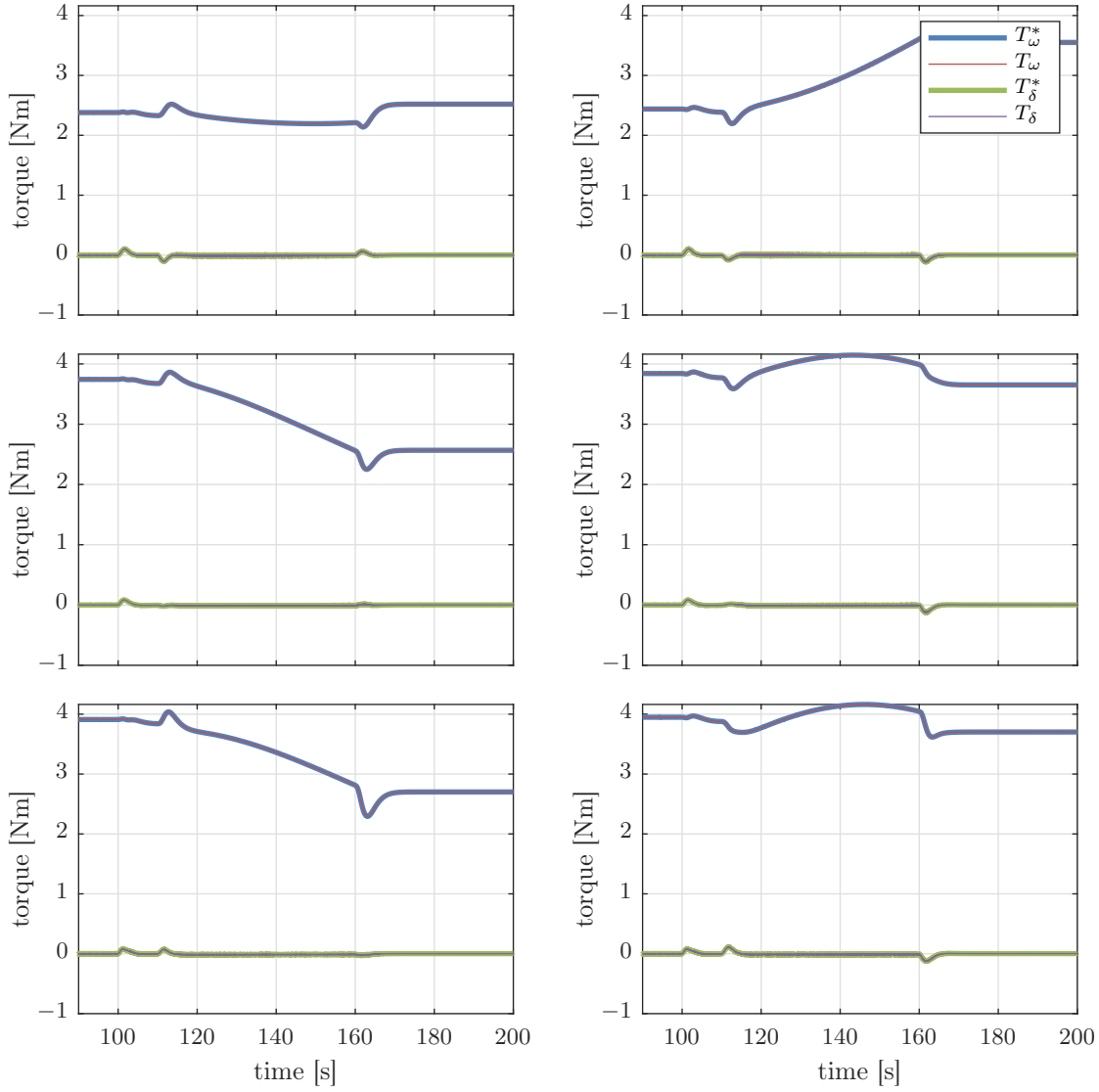


Figure 5.7: Drive and steering torque trajectories and measured values.

Wheel level. Lastly, the trajectory following behavior on wheel level, i.e. the performance of the wheel-ground feedback linearization sub-controller, is investigated. The overall quality of this sub-controller can be seen by comparing the wheel force trajectories $\boldsymbol{y}^*_{\tilde{\mathrm{W}}_i}$ (bold lines) to the achieved wheel forces $\boldsymbol{y}_{\tilde{\mathrm{W}}_i}$ (thin green and purple lines) in Figure 5.5. While the shape of the forces is followed nicely, a steady state error of varying magnitude can be observed for the forces. This error stems from inaccuracies in the quasi-static normal force estimation. It was explained in Section 3.7.2 that dynamic effects and traction forces are ignored in the force- and torque balances of the estimation. Furthermore, no error integral is implemented in the wheel level controller, which is why this steady-state error remains on wheel level and only the superordinate rover level controller compensates it with its rover velocity error integration.

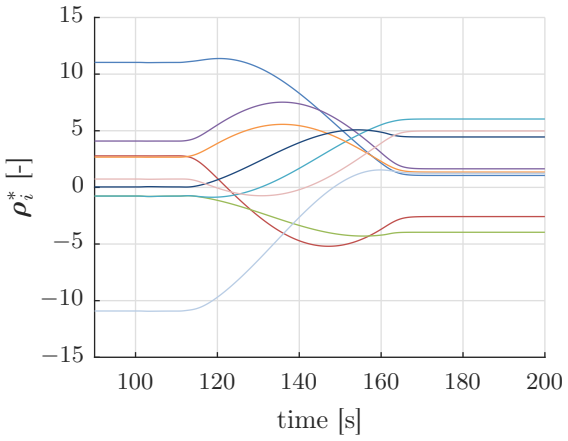


Figure 5.8: Optimized DoFs of ATC2.

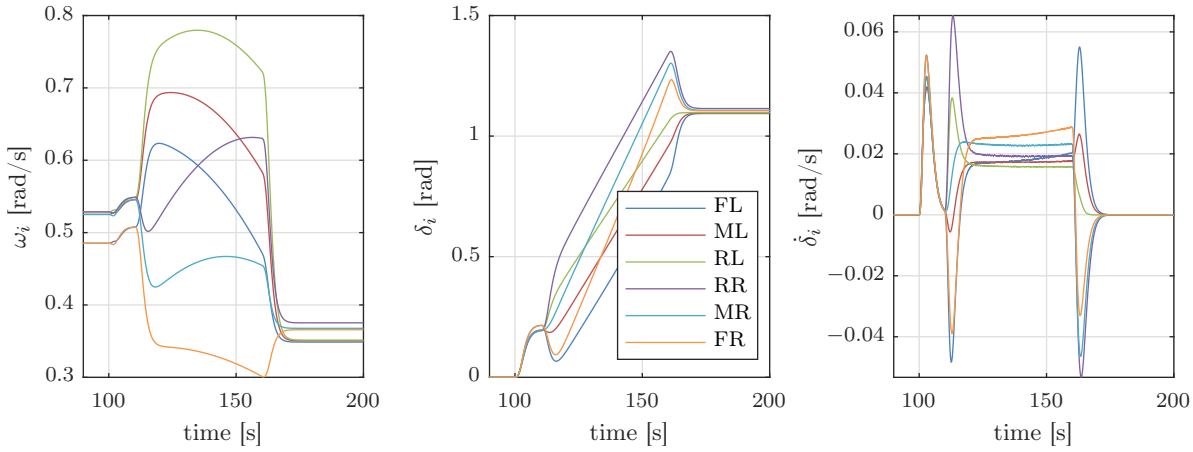


Figure 5.9: Wheel states of the ATC2-controlled rover.

The ultimate manipulation variables of the ATC2 chassis controller are the drive and steering torques, which are plotted in Figure 5.7 with the bold lines being the trajectories and the thin lines the measured values from the RST simulation. A perfect fit of measured and desired torques can be observed, which is not surprising since the actuators are modeled ideal with a linearly decreasing maximum torque, see Table 5.3. The desired values of this nominal scenario are within the limits and can therefore be achieved perfectly.

In Figure 5.9, the resulting measured wheel states are plotted (left to right): The wheel rates ω_i, the steering angles δ_i and the steering angle rates $\dot{\delta}_i$ with $i = 1 \dots 6$. Looking at the wheel rates for no commanded yaw velocity of the rover, i.e. $t \leq 100\,\mathrm{s}$ and $t \geq 170\,\mathrm{s}$,

Table 5.3: Assumed maximum rate and torque of the drive and steering actuators.

Actuator	Maximum rate	Maximum torque
drive	$1\,\frac{\mathrm{rad}}{\mathrm{s}}$	$30\,\mathrm{N\,m}$
steering	$0.24\,\frac{\mathrm{rad}}{\mathrm{s}}$	$24\,\mathrm{N\,m}$

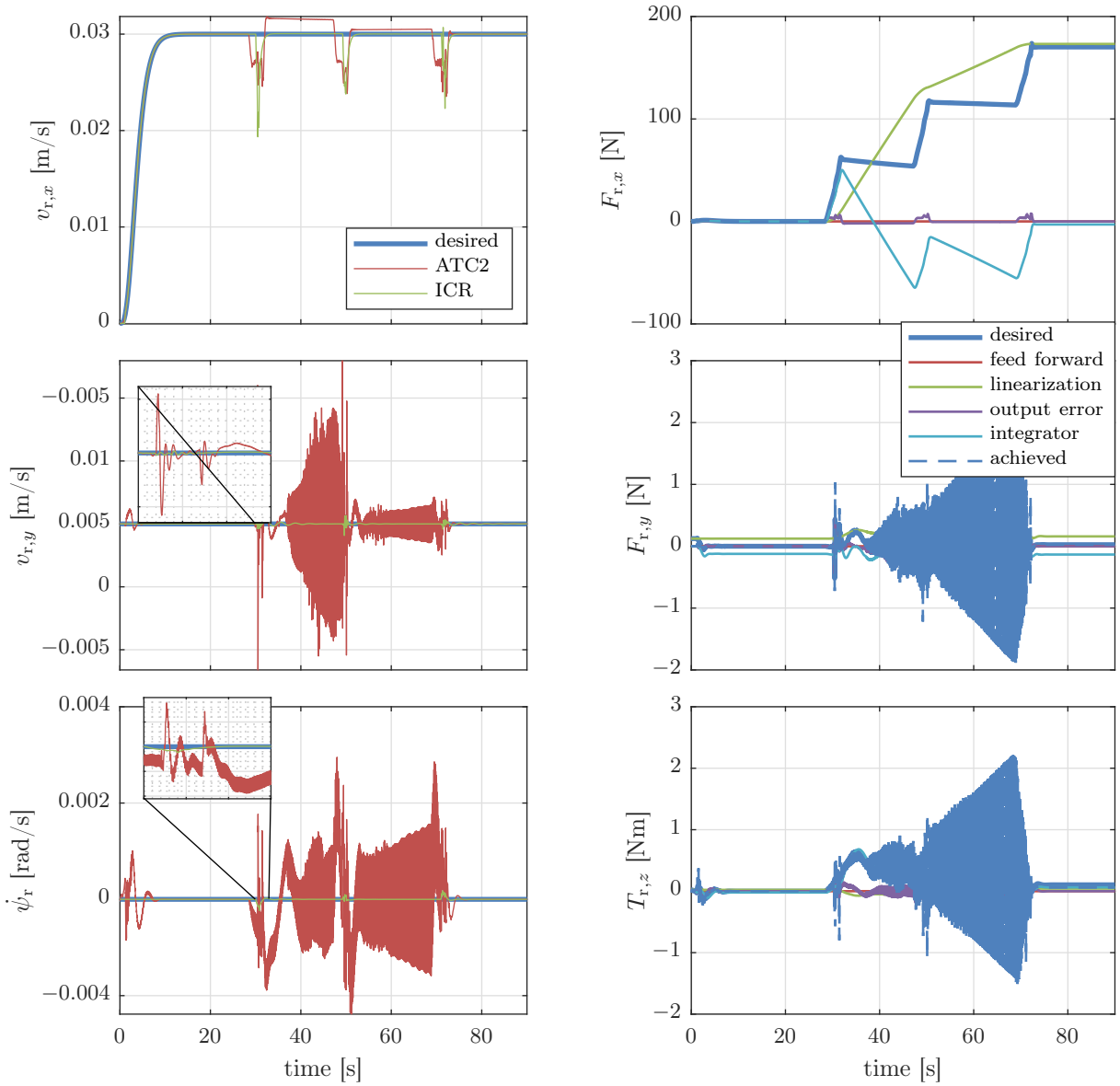
the differences between the wheels are the cause for the differing traction potential usage that was shown in the left plot of Figure 5.6. Between 100 s-110 s, the rover is commanded an additional lateral velocity component, leading to non-zero steering angles and increased wheel rates. During the turn at 110 s-170 s, the needed torque for this maneuver leads to largely differing wheel rates and steering angles between the wheels. The last part of the scenario from 170 s on, the drive that is almost orthogonal to the inclination, leads to side slippage due to the used contact model. This is counteracted by the controller with increased $y^*_{\mathrm{w}_i,y}$ forces, see Figure 5.5, which ultimately leads to considerable steering angles of about 63° for all six wheels.

5.3 Controller Robustness

In the last section, the controller performance was verified in a co-simulation with a detailed rover model. However, two of the most impactful details were nominal, i.e. as modeled in the synthesis of the controller. These two aspects are the contact angle, cf. Figure 1.5a, and the wheel-ground contact force model, which both severely influence the force generation at the contact points. Although model enhancements are certainly possible and should be investigated, as will be discussed in Section 6.2, the geometric and parametric contact situation for a rover is always highly uncertain and an exact prediction is not even possible off-board with an extended sensor suite today. To cope with these uncertainties and model errors, it is important for a rover chassis controller to be robust against them. This section therefore contains a robustness investigation of the proposed ATC2's performance to both of the above mentioned ground contact model errors. For the disturbance of non-zero contact angles, the first 90 s of the scenario from Section 5.2.2 are investigated in Section 5.3.1. The rover driving over the kink yields abruptly and largely changing contact angles for the wheels. Secondly, the ground geometry is enhanced with a realistic unevenness such that the contact angles are prevalently non-zero in Section 5.3.2. Additionally, the much more complex, discretized Soil Contact Model (SCM) is chosen for the co-simulation for more realistic contact force computation.

5.3.1 Nonzero Contact Angles

For the assessment of the controller performance for non-zero contact angles, the first part (0 s-90 s) of the scenario that was shown in Figure 5.2 is used. The rover velocity trajectories and measured velocities are plotted in Figure 5.10a, with the same scheme as in the last section: The plots show forward, sideways and yaw velocity from top to bottom and the blue, bold line is the trajectory, red is ATC2 and green the reference ICR controller. Within this first part, there are two challenging periods for the controller, the first is from the start until approximately 5 s, where the rover is almost standing still and the desired forward velocity is only slowly built up. The second part is the section from 30 s to approximately 70 s where the front, middle and rear wheel pair reach the kink from 0° to 10° one after the other.



(a) Velocities (b) Forces/Torque

Figure 5.10: Rover velocity and force trajectory following for non-zero contact angles with the nominal contact model.

When a wheel drives over that position of the surface, the contact plane, in which normal and tangential forces are calculated in the RST, tilts and is no longer perpendicular to the steering axis. Due to the zero contact angle assumption that was made for the controller synthesis, this implies a disturbance. Not only the directions of the tangential forces but also the direction of the normal force tilts with the plane, which has a significant influence on the rover movement. That normal force is large compared to the tangential forces, particularly when the first wheel hits the kink since there is almost no traction force needed for the drive on the plane as shown in the first plot of Figure 5.10b at 30 s. The tilt of the normal force vector is opposed to the drive direction of the rover and needs to be compensated for by

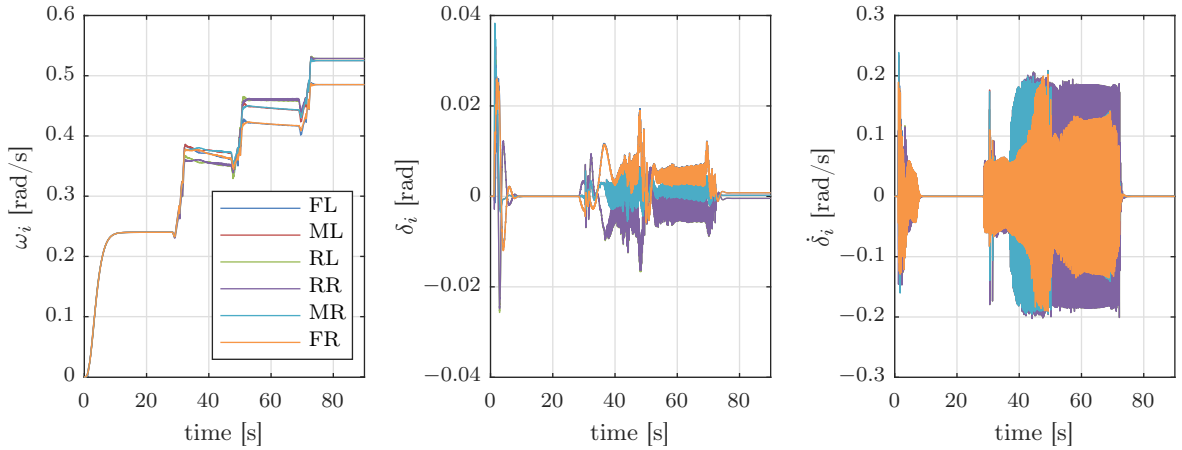


Figure 5.11: Wheel states for non-zero contact angles.

the velocity error integrator, which is the cyan plot in Figure 5.10b. Looking at the cyan and green curves, it can be seen how the integrator first needs to cover the necessary uphill force and the model-based linearization term slowly takes over as the measured rover pitch angle rises. Once the rover is on the inclination with all six wheels at about 70 s, the rover orientation matches the slope, all contact angles go back to zero and so does the velocity error (purple) and its integrator (cyan). This disturbance, however, does influence the control performance of the lateral and yaw direction considerably as can be seen in the shuttling in the bottom two plots of Figure 5.10a. For comparison, the reference ICR controller is plotted as well and it is very clear in the magnifications of the plots that it suffers much less than ATC2 from the large change in needed traction forces. The controller is not able to converge to the lateral and yaw velocity trajectories during the non-zero contact angles, as can be seen in the oscillations. Looking at the y-axis scales of Figures 5.10a and 5.10b, it can be seen that the force oscillations stay in a range of few Newton / Newtonmeter and the velocity oscillation magnitude is about one order of magnitude below the desired forward velocity.

For understanding where this problem arises, the desired and achieved wheel torques are shown in Figure 5.12. While the trajectory always stayed below the torque limits in the nominal case in Section 5.2.3, the desired steering torque (green curve in Figure 5.12) exceeds these limits by far. The actual steering torques (purple) are, however, limited to 24 N m in the RST model and hence cannot follow. Ultimately, the resulting wheel states are plotted in Figure 5.11 to show what these oscillations lead to in terms of the actual wheel rate and steering angles. The steering angle deviation due to the disturbance stays below 6° and the high frequency oscillations between 40 s and 70 s have an amplitude of about 0.35°.

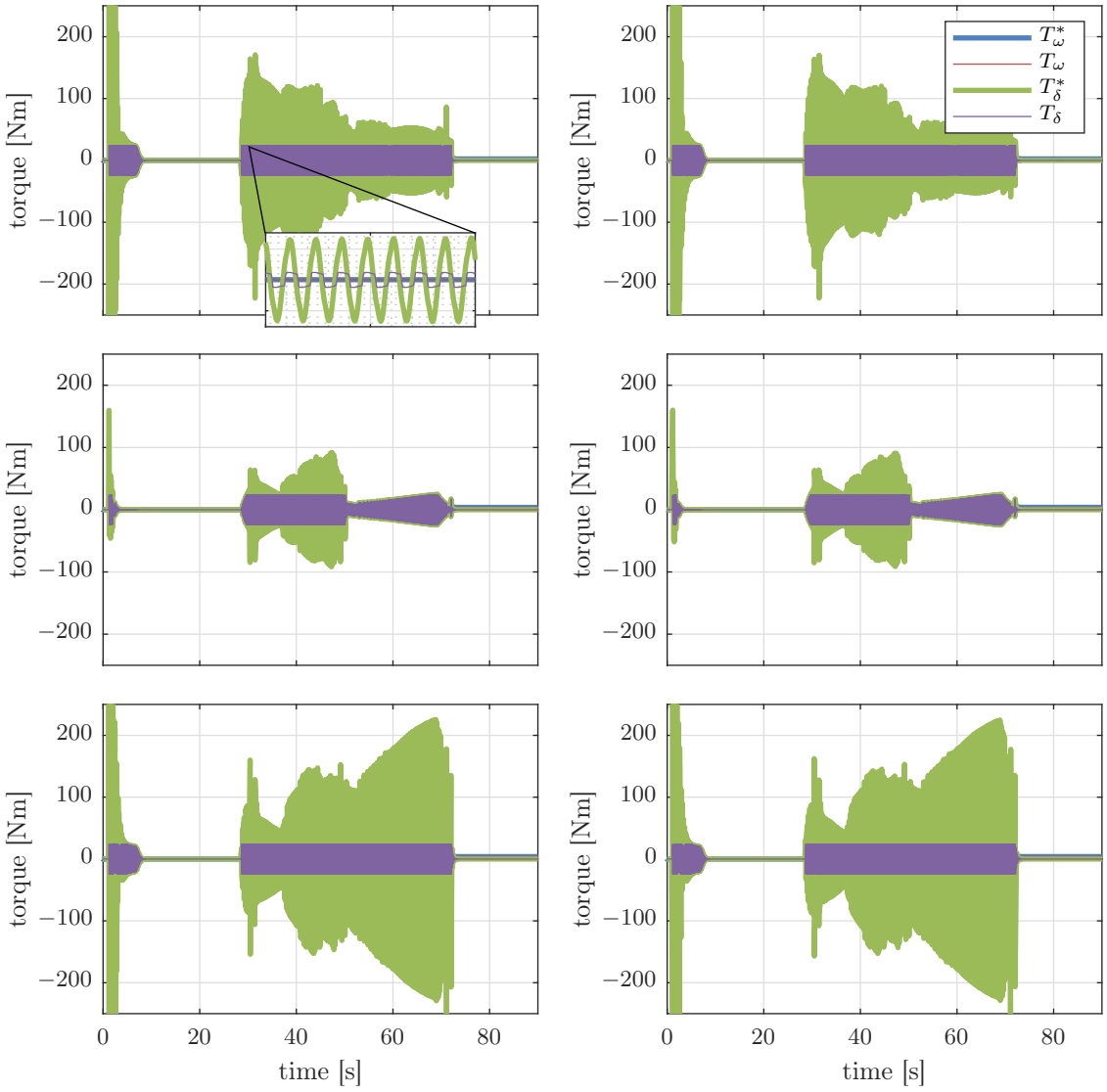


Figure 5.12: Wheel torques for non-zero contact angles.

5.3.2 High-fidelity Contact Model and Soil Unevenness

In this section, the proposed controller is tested with the high-fidelity SCM for the wheel-ground contact and an uneven terrain. The contact model was briefly mentioned in Sections 1.3 and 3.5: It is based on an equally distributed 2-dimensional mesh of the ground surface geometry. Each node of this mesh has several states, which are computed by the two modules of SCM: force-/torque computation and the soil flow algorithm. As the name suggests, the latter is responsible for calculating the deformation of the sandy ground, which includes a soil height at each node as well as a soil flow depth and velocity. It is easily understood that the geometric deformation, but also the other soil states, have a major influence on the contact force calculation. Such a complex model cannot be integrated into

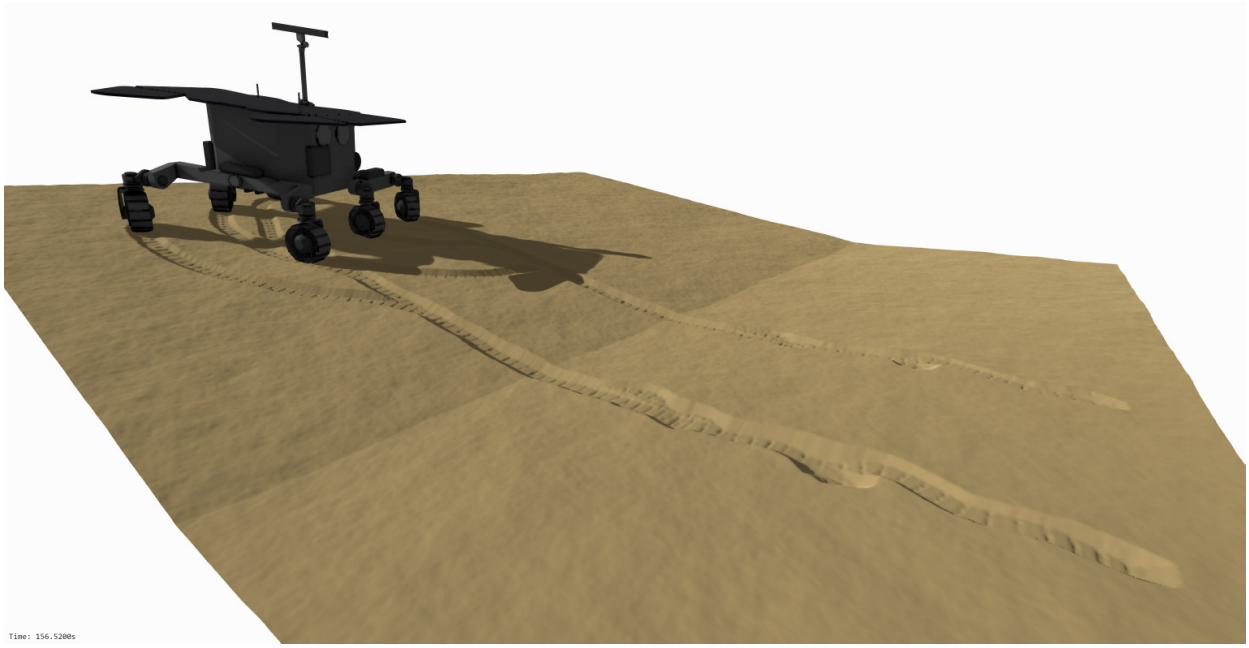
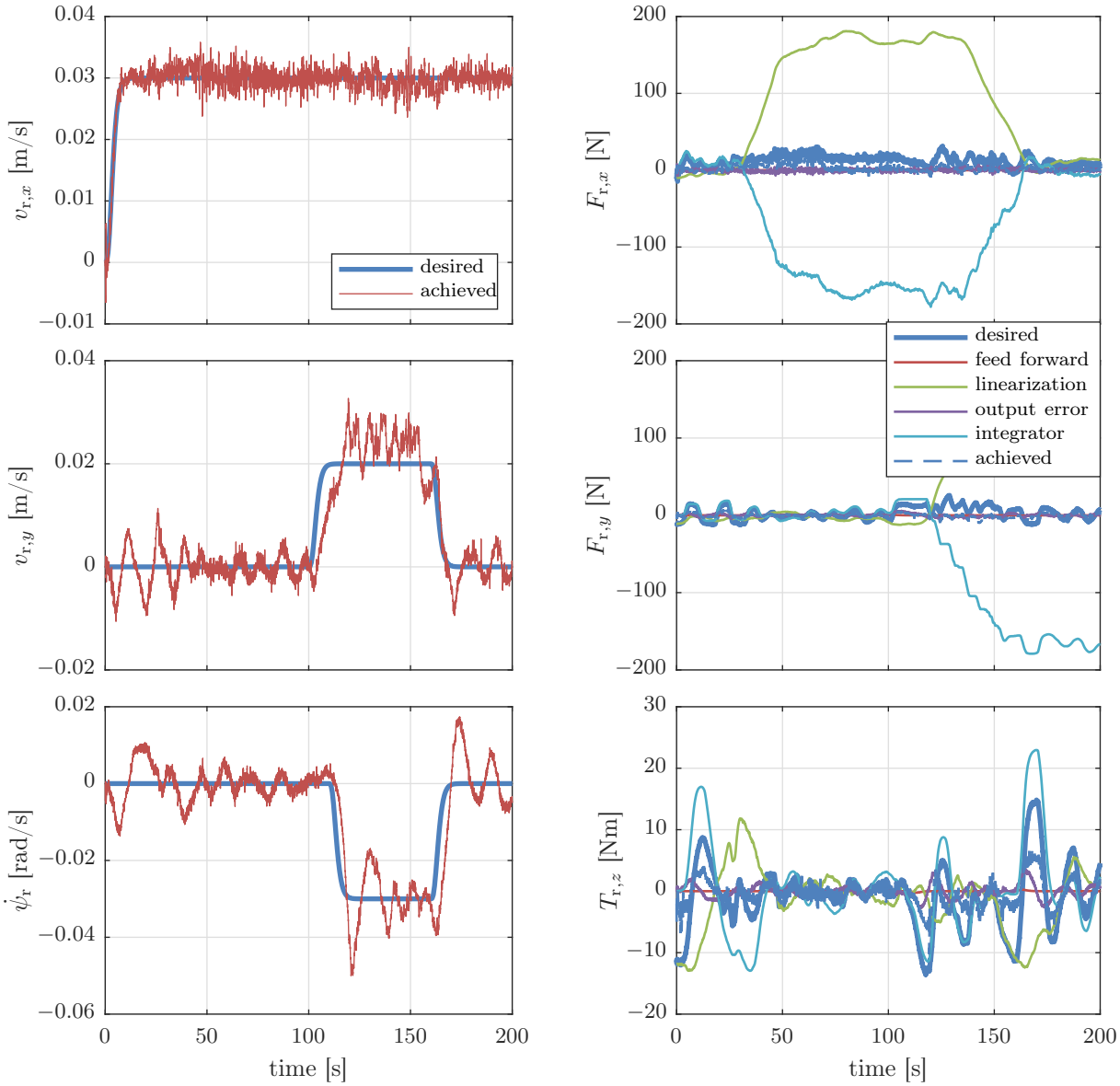


Figure 5.13: Screenshot of the ExoMars rover with the SCM contact model during the turn on the inclination (with our in-house 3D visualization [HBS14]).

a controller synthesis directly due to its structure and computational effort, however, the controller should be able to handle this more realistic contact model and the large disturbances that are associated with it. ATC2 should hence be able to follow the desired velocity trajectory despite the model errors and disturbances.

Additionally, actual ground is never completely flat, even without larger obstacles like rocks there is an unevenness which can be replicated with the superposition of multiple noise functions. This feature is included in the soil generator of the RST and the chosen surface can be seen in Figure 5.13, where also wheel ruts due to SCM's soil deformation algorithm appear. The image is a screenshot from the *DLR Visualization Library* [HBS14], a 3D visualization that is attached to Modelica models and is fully included in the RST.

Figure 5.14a shows the considerable impact of this non-nominal contact force model and the non-zero contact angles of the uneven ground in terms of the rover velocity trajectory following performance. Three major effects can be seen, the first one being a high frequency noise that is especially prominent in the forward velocity (first plot in Figure 5.14a). This forward velocity is, on average, very close to the desired one, however, it shows a noise-type disturbance with a magnitude of about 10-15 % of the average value. The causes of this disturbance is mainly the used SCM contact model, which produces noisy forces even for steady state movement due to its discretization of the soil. Additionally, the model considers the effects of the grousers on the wheels, therefore there is always an oscillation in the forces, caused by the grousers entering and leaving the soil contact. Although the discretization effect is numerical in nature, such random noise-type disturbances occur in reality as well, initiated by the highly noncontinuous contact between a wheel and sand, consisting of individual particles. Secondly, there are considerable low-frequency deviations



(a) Velocities (b) Forces/Torque

Figure 5.14: Rover velocity and force trajectory following for the non-nominal SCM contact model and uneven ground.

of the sideways and yaw velocity from their trajectories in the middle and bottom plot of Figure 5.14a. In Figure 5.13, the results of these deviations can be seen by means of the non-straight wheel ruts at the beginning. These errors, especially for the start of the rover, are the result of the non-modeled resistance forces of the soil in combination with the actuator limits. The large steering angle oscillations can be seen in the middle plot of Figure 5.15. Although these oscillations get smaller once the rover moves, the non-considered steering actuator torque limits and ongoing disturbances prevent full convergence. In this scenario, one of the steering actuators even does two full rotations (orange plot) before it settles to an angle again. Due to the affected wheel being almost in the center of the rotation,

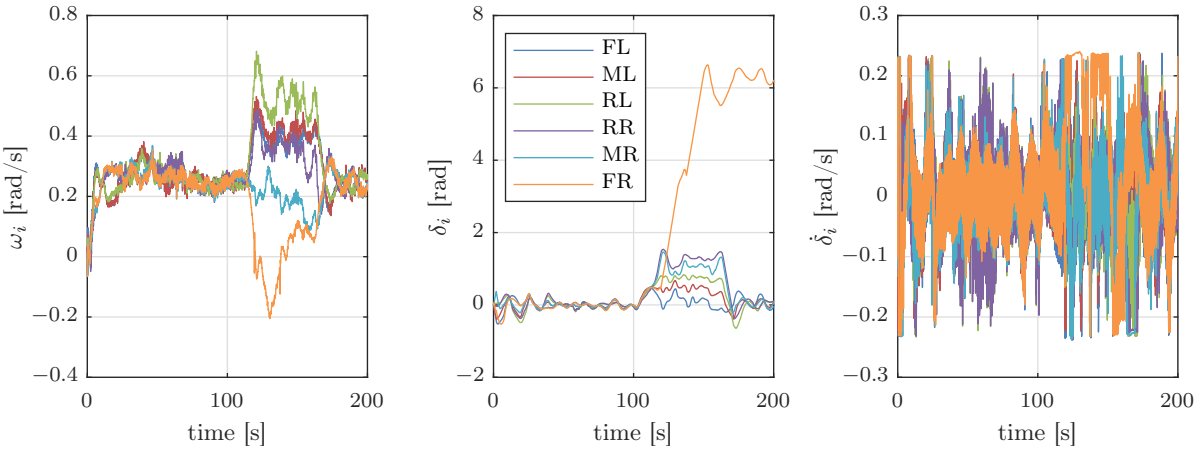


Figure 5.15: Wheel states for the non-nominal SCM contact model and uneven ground.

it does not get a clear force direction, thus the disturbances predominate for that wheel. The third effect is a mean value deviation of the average forward and, mainly, sideways velocities for the curve drive maneuver of the rover between 120 s and 160 s in Figure 5.14a. Looking at the rover force and torque plots in Figure 5.14b, the linearization (green) and the error integrator (light blue) contribute most to the total (blue) force trajectory in forward and sideways direction. In fact they almost cancel each other on the inclination in forward direction between 40 s and 170 s and sideways from 120 s on. It needs to be said that the achieved forces in the plot are based on the nominal contact model calculations and are not the actual contact forces of SCM. However, the desired force yields, through the contact model of ATC2, to similar wheel rates and therefore similar slip with and without inclination. This can be clearly seen by comparison of the wheel rates in the period 20 s-100 s, which are shown in Figure 5.15 for SCM and Figure 5.11 for the nominal contact model. The latter shows distinct steps in the wheel rates at 30 s, 49 s and 70 s which are the times where the front, middle and rear wheel pairs hit the kink, respectively. With SCM on the other side, no correlation between the wheels driving over the kink and the required wheel rates can be seen which ultimately means that in this parameterization of SCM, the rover drives with similar wheel slip on the plane and a 10° inclination. In terms of the proposed controller, this is a considerable model error, which needs to be compensated for by the integrator. The implications for the trajectory following are similar to the elaborations about the downhill force compensation of the ICR controller towards the end of the *wheel level* paragraph in Section 5.2.3. A rapid change in directions cannot be followed adequately due to the slow behavior of the integrator. It essentially needs to shift from x- to y-direction during the ~90° turn between 110 s and 170 s, which is the cause for the mean value error in the forward and sideways velocities.

5.4 Conclusion and Discussion of the Results

A verification of the controller performance under nominal conditions, as well as with disturbances and model errors, was undertaken in this chapter. To conclude the simulative investigation, the findings are summarized and discussed in this section.

The nominal contact model and zero contact angles, i.e. as assumed for the controller synthesis, were used in the rover simulation and the ATC2 controller was compared to a geometric constraint-based ICR reference controller in Section 5.2.3. This is referred to as the nominal case, although model errors between the synthesis model and the RST co-simulation model exist. These main discrepancies influence the shown results in Section 5.2.3 and are summarized in the following list:

- Mass and inertia distribution: In the synthesis model, the rover dynamics are simplified to rigid body dynamics, i.e. all masses are assumed in one rigid box around the CoG and a wheel inertia at the actuators. Especially the inertia that is caused by bogie and wheel assembly masses is neglected. These simplifications lead to an inaccuracy in both, the CoG position and the dynamics, especially of the bogie rotation.
- Normal contact force estimation: This estimation is described in Section 3.7.2 and is based on quasi-static force-torque balances that only take forces in the direction of the gravity into account. Especially growing traction forces, e.g. on an inclination, introduce considerable error into this estimation.
- Actuator torque and contact force limits: In the synthesis, no torque limits were considered, which is in contrast to the rate-dependent maximum torque that is implemented in the evaluation RST model. Additionally, the maximum contact forces, which limit the effector output, are not considered explicitly in the allocation, the rigid body dynamics or the trajectory generation. For fast maneuvers or large errors, the controller could thus command contact forces and/or actuator torques that are not feasible and might, in consequence, not yield the optimal solution due to that lacking knowledge of the limits.

For the shown scenario, which includes driving straight, sideways velocity and a turn on a 10° inclination, the overall trajectory following errors were small. Compared to the reference controller, minor improvements in the velocity trajectory following performance could be seen. A large difference was revealed in the needed integrator action, which is very small for the ATC2 controller in contrast to the ICR controller, due to lacking model knowledge of the latter. Of the model inaccuracies summarized above, the normal force estimation error was most prominent in that the actual wheel forces showed an offset to their desired values. Although the optimization based force allocation was able to keep the traction potential usage difference of the wheels lower than the reference controller during the turn, it showed differences of up to 6 % between the wheels. Whether this is purely due to the modeling

errors from above or the nonlinear optimization is not able to find a global minimum should be investigated in future work.

For testing the robustness of the proposed controller to larger disturbances, a straight drive over a kink in the ground geometry from an inclination of 0° to 10° was investigated in Section 5.3.1. This kink leads to non-zero contact angles at the wheels which is a disturbance compared to the assumed contact angles of zero. The contact angle error introduces an error in the assumed direction of the contact forces, the largest one being the normal force. As expected, the error integrator needs to compensate for this error in the rover's forward direction until the rover is fully on the inclination and the contact angles are back to zero. Besides these expected errors, oscillations in the rover sideways and yaw direction were observed during the phase of non-zero contact angles, which are mainly initiated by small asymmetries of the contact force direction on the left and right side. The oscillation is then supported by the lacking consideration of limits in the controller: The ATC2 controller commands torques that are not feasible in this disturbed period. For the cascaded sub-controllers it was, however, assumed that the underlying controller(s) are always able to achieve the trajectory. The discrepancies between actual and desired torque of the steering actuator thus lead to errors in the achieved wheel forces and thereby also the overall rover forces. While this results in relatively small steering angle oscillations of smaller than 0.4° and maximum peaks of about 6° steering angles and hardly compromises the overall trajectory following performance, it might have negative effects e.g. on energy consumption and actuator lifetime.

With the same drive scenario, however, using an uneven soil geometry and the 2,5D wheel-ground interaction model SCM, a more realistic simulation was chosen to evaluate the controller's performance for larger disturbances. Since it is not possible to consider such a complex contact model in the controller synthesis, effects of the modeling errors onto the controller performance with its much simpler synthesis model were studied in Section 5.3.2. Three major effects of the large model errors and non-zero contact angle disturbances were found: First, a high-frequency noise-type deviation of the velocities from their trajectories due to grouser and discretization effects from the SCM contact model and its spatial discretization. Second, low-frequency deviations of the sideways velocity and yaw rate mainly result from sinkage-dependent, varying resistance forces and the steering actuator torque limits that are not considered by the controller. Last, an offset in the average values of the forward and sideways velocity during the turn results from large error integrator actions that only adjust slowly to the changing downhill force during the turn of the rover on the inclination. This large integrator value results from much lower required downhill force compensation with the SCM model compared to the nominal contact model. Assuming that the higher-fidelity SCM model is correct in its force computation, the synthesis model should be corrected. A novel fast terramechanics model was developed in Section 3.6, however, it is not directly usable in the controller synthesis, see Section 3.6.4 and Section 6.2 for future work.

Overall, it was shown in this chapter that the trajectory following behavior in the nominal case is good and it shows the expected advantages of a model-based controller and the optimal force allocation. The robustness tests showed that the controller is able to follow the desired velocity trajectory even for considerable disturbances and model errors. Undesired effects of disturbances and model errors were shown, and their causes were explained. Generally, the trajectory following is achieved even for larger disturbances, however, disturbances like soil unevenness or sensor noise lead to relatively large force compensations by the controller. This is due to the 3-dimensional rigid body model taking the normal forces, which are large compared to the traction forces, into account. The 3-dimensional compensation allows the shown low integrator action for driving the rover on inclinations but due to the described sensitivity to disturbances, it is also responsible for a somewhat nervous behavior of the controller in the robustness tests. This topic for further investigations is picked up again in Section 6.2.

6 Conclusion and Outlook

6.1 Conclusion

A fully model-based chassis controller for an over-actuated planetary exploration rover has been developed in this work. Thereby, two topics were in focus: Firstly, the consideration of wheel-ground models in the controller. Secondly, the explicit computation of the manipulation variables albeit exploiting the full solution space for the optimization of the available DoFs. The controller has been verified in co-simulation with a high-fidelity rover and terramechanics model and advantages and existing problems were shown.

The thesis started with a motivation for the topic, in which difficulties of rovers driving on soft sand were summarized. It was shown in a structured literature review how most chassis controllers don't use the available DoFs to optimize the allocation but rather introduce additional constraints. Based on the understanding that the wheel-ground contact forces and terrain inclinations play a crucial role in the locomotion of a rover, the decision for a fully model-based controller with an allocation on force level was motivated. For the first part of the modeling, common approaches from robotics were chosen. The direct kinematics was derived by defining coordinate systems for each joint of the parallel kinematic chains and applying the DH systematic. These kinematics needed to be completed with the moving base of the rover and its partial derivative. The geometric Jacobian was used to state the relative kinematics, which represents the relation between rover and ground contact point velocities. By applying the kineto-statics duality, the force-torque sum of contact point and rover CoG forces as well as torques was obtained. Newton's law in three dimensions was used for a simplified rigid body to derive the rover-level dynamics under consideration of gravitational forces on inclined terrain. On single wheel level, the dynamics of drive and steering actuator were stated under the assumption of an ideally realizable torque. Before the synthesis ground contact model could be addressed, a brief introduction into semi-empirical terramechanics models was provided, including a discussion of their problems when it comes to the use in onboard control software. Additionally, a novel empirical, dynamic terramechanics model for time-critical applications was developed and the challenges and open questions for using it in the chassis controller were raised. A one-point contact force model with a slip-velocity based traction force computation and a lateral force were finally derived and integrated into the wheel dynamics to form a combined wheel-ground dynamics synthesis model. This model provides a relation between the actuator torques as inputs and the individual contact point forces, that act on the rover, as outputs. More complex ground models with dynamic states can thereby easily be added in later work.

In terms of designing a controller for a dynamic, over-actuated system, an existing, general approach was presented for dividing the full system into controllable dynamic systems with unique solutions and a static, linear underdetermined system of equations. The advantage of this separation into sub-models, and sub-controllers respectively, is that the dynamic systems can be handled with well-known control approaches for determined systems, while treating the over-actuation as a simple, linear system. For the latter, the general solution consists of the least-squares solution, which is computed through the pseudo-inverse, and the homogeneous solution, which uses a base of the null-space of the allocation matrix. This provides a possibility to combine the advantages of an explicit, fast computation of the manipulation variables and an exploitation of the full solution space. To be able to apply this structure, each of the sub-models' outputs has to be controllable and it must be ensured that the underlying sub-controllers within the cascaded structure are able to achieve the computed trajectories of the preceding sub-controller. One main contribution of this work, therefore, is to modify the derived sub-models such that these necessary properties are met, which was described in detail. Firstly, the effector states were ensured to only appear in the effector dynamics to enable the controller separation. Secondly, the bogie-angle dependent allocation matrix was analyzed for rank deficiency and modified accordingly. Thirdly, identified non-controllable parts were added to the rover-level dynamics model to ensure completeness and correctness of the full model. A feedback linearizing controller was designed for the nonlinear rigid body dynamics model. Due to sum of the relative degrees being less than the system order, the closed-loop system has unobservable internal dynamics. Their stability was analyzed with Lyapunov's indirect method, which is an analysis of the eigenvalues of the linearized zero dynamics. This analysis was achieved by the choice of a different contact model that considers the position-dependency of the normal forces. For the allocation, the above mentioned separation was derived with the pseudo-inverse and a null-space base of the force-torque allocation matrix and a nonlinear objective function was presented for a minimization of the soil failure risk. The underlying wheel-ground dynamics could not be feedback linearized directly due to a singularity of the decoupling matrix. To solve that problem, a dynamic extension was performed by using the derivative of the drive torque as a new input instead of the torque itself. By doing so, the relative degree was increased and a feedback linearization became feasible. However, one of the measurable disturbances could not be assumed piecewise constant and therefore needed to be considered in the time derivative of the subsystem outputs. The partial derivative with respect to this disturbance was added to the one with respect to the states. Short sections at the end of each chapter summarized the model development with block diagrams of both model and controller.

A simulative verification was done with a co-simulation of the controller in MATLAB/ Simulink and a high-fidelity rover model in the multi-physics modeling language Modelica. For comparison of the proposed controller, a state of the art controller, based on the Instantaneous Center of Rotation (ICR), was implemented. The performances were compared for the nominal case, i.e. with small disturbances and with using the synthesis contact model in the high-fidelity model as well. Robustness investigations were presented for non-zero

contact angles as well as a high-fidelity, discretized soil contact model that includes soil deformation and soil flow computation. Overall, a good trajectory following behavior was achieved with the proposed controller including the expected benefits of the fully model-based approach and the soil failure risk minimization. The controller was able to not only follow the trajectory in the nominal case with minimal error or integrator action, it was also able to keep the soil failure risks within 6 % difference across all wheels, which is an improvement of about 10 % compared to the least-squares solution. It was also shown that an optimization only every 100 controller time steps did not compromise the quality, which is important for keeping an optimization-based controller algorithm computationally efficient. Observed oscillations in the disturbed cases were mainly caused by not considering the contact force and actuator limits in the controller and by taking the full 3-dimensional gravitational forces into account. The latter enables the gravity force compensation but also yields large disturbance forces for relatively small disturbance angles in the contact geometry. While this causes a noise-type disturbance as well as a nervous behavior, the controller was always able to achieve a decent velocity trajectory following, even for relatively large disturbances and model errors.

6.2 Outlook

Possible topics for further research, that arise from the observed problems or further goals of the rover chassis control, are summarized in this section.

It was pointed out several times throughout the thesis and also observed in the simulative results that the wheel-ground contact model is crucial for the considered system. While a simple, fast and robust model was found and used in this work, the range of validity for different sands and conditions was not fully investigated. A novel model approach towards a fast, yet accurate terramechanics model has been developed in Section 3.6, which, however, cannot be integrated in the controller synthesis yet, as explained in Section 3.6.4. Further work that needs to be done in terms of this model is summarized in [Bar18]. When it comes to experimental validation, there is a lack of single wheel test campaigns that consider the dynamic behavior of the wheel-ground contact forces, e.g. for a wheel that stops, starts over, reverses, accelerates, drives through a rut of the preceding wheel or is dragged through the soil. A thorough investigation of lateral forces and gravitational effects on slopes is needed, but not known either. Preparations for dynamic single wheel tests were done and a first small-scale test campaign for some of the dynamic effects above was pursued at our institute and our Terramechanics Robotics Locomotion Lab [BBL18] enables large testing flexibility and automation. Another crucial model assumption was the contact angle of zero, which can become incorrect to different degree for different driving situations. Small contact angles result from mildly uneven terrain and higher ones e.g. for obstacle crossing. The most extensive work in the area of contact angle estimation was done by Iagnemma and Dubowsky [ID04a]. Some investigations were done at our institute as well in combination with a velocity estimation purely from IMU, bogie and steering angle and wheel rate

measurements [Vij18]. It was furthermore shown that the quasi-stationary estimation of the normal contact forces yields deviations which increase for increasing tangential forces. An integration of the tangential forces into the force torque sum could be investigated in future work.

Besides the terramechanics model and contact angle estimation themselves, it is another subject for research how these could be integrated into the controller synthesis of the rover. The contact angles were assumed to be zero in the derivation of the rover kinematics, which allowed to state these kinematics without a steering angle dependency. An analysis would need to show, if the proposed controller layout is still possible with non-zero contact angles. Different possibilities arise for the integration of a more complex terramechanics model. As described in Section 3.6.4, it is not directly possible to use it in the existing wheel-ground dynamics and still achieve the necessary continuous differentiability for the controller synthesis. Therefore, the feasibility of a reduced model that fulfills the requirements could be studied. As an alternative, the developed terramechanics model could be used outside the main control loop for an estimation of model parameters onboard, which are then used in terms of a parameter adaptive controller. Large differences for the contact forces on different soil types could also require to switch the entire controller structure when a certain ground contact situation is detected.

The results also suggested that taking the contact force and actuator limits into account could improve controller performance. Actuator limits were e.g. addressed in the control allocation survey of Johansen and Fossen [JF13], where the limits are mostly used as constraints in the optimization. The force limits could be precised with further analysis of the output controllability region of the wheel-ground dynamics and taken into account as well. For this controller, it was found that enforcing the nonlinear constraints in the optimization makes it difficult to solve in some situations and also might require to perform it more often than every 100th controller time step. If possible, it would certainly be more efficient to consider the limits already in the trajectory generation. A study about the feasibility of the optimization, the existence of a global optimum and the ability to find it are further possible research topics.

A torque control of each actuator is assumed in this work. During tests with the actual drive trains of the ExoMars BB2 prototype, it was, however, found to be difficult for the servo drives in certain situations to achieve the desired torque and control it without a bang-bang behavior between the rate limits. It could thus be investigated if the wheel-ground dynamics can be stated with a wheel rate and steering angle input instead of the torques and what impact this would have on the controller synthesis. A synthesis might or might not be feasible with the different manipulation variables depending on the used terramechanics model and could simplify the consideration of actuator limits in the optimization, that was discussed above. The desired force-based allocation should be feasible irrespectively of the used manipulation variables for the motors.

It was shown in the results that the full normal force consideration yields a certain nervousness of the controller in case of large disturbances. A different consideration of the downhill force effects in the 3-dimensional rigid body dynamics could be investigated. Future, faster rovers might furthermore require an extension of the dynamics to full multi-body dynamics.

Finally, due to their remote field of application, an important capability for rovers is Failure Detection, Isolation and Recovery (FDIR). Due to the actuator redundancy, rovers are usually able to continue driving even with an actuator failure. The applicability of the optimization-based chassis controller of this work to such situations could be investigated.

Literature

[AA18] Hamzah M Beakawi Al-Hashemi and Omar S Baghabra Al-Amoudi. „A review on the angle of repose of granular materials“. In: *Powder technology* 330 (2018), pp. 397–417.

[ABW19] DC Agle, Dwayne Brown, and JoAnna Wendel. *NASA's Opportunity Rover Mission on Mars Comes to End*. [press release]. National Aeronautics and Space Administration (NASA), Feb. 13, 2019. URL: `https://mars.nasa.gov/news/8413/nasas-opportunity-rover-mission-on-mars-comes-to-end/` (visited on 07/19/2019).

[Ada14] Jürgen Adamy. *Nichtlineare Systeme und Regelungen*. Springer, 2014. ISBN: ISBN 978-3-662-55684-9.

[Agl19] DC Agle. *Mars 2020 Rover: T-Minus One Year and Counting*. [press release]. National Aeronautics and Space Administration (NASA), July 17, 2019. URL: `https://mars.nasa.gov/news/8461/mars-2020-rover-t-minus-one-year-and-counting/` (visited on 07/24/2019).

[AKB06] Johan Andreasson, Christian Knobel, and Tilman Buente. „On Road Vehicle Motion Control – striving towards synergy“. In: *Proceedings of the 8th International Symposium on Advanced Vehicle Control (AVEC)*. 2006, pp. 20–24.

[All97] Michael Allison. „Accurate analytic representations of solar time and seasons on Mars with applications to the Pathfinder/Surveyor missions“. In: *Geophysical Research Letters* 24.16 (1997), pp. 1967–1970.

[Apf+11] Maximilian Apfelbeck et al. „A systematic approach to reliably characterize soils based on Bevameter testing“. In: *Journal of Terramechanics* 48.5 (2011), pp. 360–371.

[Arv+10] Raymond E Arvidson et al. „Spirit Mars Rover Mission: Overview and selected results from the northern Home Plate Winter Haven to the side of Scamander crater“. In: *Journal of Geophysical Research: Planets* 115.E7 (2010).

[Arv+16] Raymond E Arvidson et al. „Mars Science Laboratory Curiosity Rover Megaripple Crossings up to Sol 710 in Gale Crater“. In: *Journal of Field Robotics* 34.3 (2016), pp. 495–518.

[BA06] Tilman Bünte and Johan Andreasson. „Integrierte Fahrwerkregelung mit minimierter Kraftschlussausnutzung auf der Basis dynamischer Inversion“. In: *VDI Berichte* 1931 (2006), p. 163.

[BBL18] Fabian Buse, Tobias Bellmann, and Roy Lichtenheldt. „The Terramechanics Robotics Locomotion Lab“. In: *Proceedings of the 14th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*. 2018.

[BDO18] Kristin Bussmann, Alexander Dietrich, and Christian Ott. „Whole-Body Impedance Control for a Planetary Rover with Robotic Arm: Theory, Control Design, and Experimental Validation“. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 910–917.

[Ber+19] Jean Bertrand et al. „Roving on Phobos: Challenges of the MMX Rover for Space Robotics“. In: *Proceedings of the 15th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*. 2019.

[BG74] Adi Ben-Israel and Thomas NE Greville. *Generalized inverses: theory and applications*. New York: Wiley, 1974. ISBN: 9780471065777.

[BM06] Jeffrey J Biesiadecki and Mark W Maimone. „The Mars Exploration Rover Surface Mobility Flight Software Driving Ambition“. In: *Proceedings of the Aerospace Conference*. IEEE. 2006, 15–pp.

[Bor+17] Leonardo Bora et al. „ExoMars Rover Control, Localisation and Path Planning in a Hazardous and High Disturbance Environment“. In: *Proceedings of the Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*. 2017.

[Bus18] Fabian Buse. „Using superposition of local soil flow fields to improve soil deformation in the DLR Soil Contact Model - SCM“. In: *Proceedings of the 5th Joint International Conference on Multibody System Dynamics (IMSD)*. 2018.

[Din+10] Liang Ding et al. „Wheel slip-sinkage and its prediction model of lunar rover“. In: *Journal of Central South University of Technology* 17 (2010), pp. 129–135.

[Din+15a] Liang Ding et al. „Improved explicit-form equations for estimating dynamic wheel sinkage and compaction resistance on deformable terrain“. In: *Mechanism and Machine Theory* 86 (2015), pp. 235–264.

[Din+15b] Liang Ding et al. „Interaction mechanics model for rigid driving wheels of planetary rovers moving on sandy terrain with consideration of multiple physical effects“. In: *Journal of Field Robotics* 32.6 (2015), pp. 827–859.

[Ell16] Alex Ellery. *Planetary Rovers - Robotic Exploration of the Solar System*. Springer, 2016. ISBN: 9783642032585.

[EMO99] Hilding Elmqvist, Sven Erik Mattsson, and Martin Otter. „Modelica-a language for physical system modeling, visualization and interaction“. In: *Proceedings of the International Symposium on Computer Aided Control System Design*. IEEE. 1999, pp. 630–639.

[Eur20] European Space Agency. *ExoMars to take off for the Red Planet in 2022*. [press release]. Mar. 12, 2020. URL: http://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/Exploration/ExoMars/ExoMars_to_take_off_for_the_Red_Planet_in_2022 (visited on 03/30/2020).

[Fou19] Jeff Foust. *SpaceIL lander crashes on moon*. [press release]. SpaceNews, Apr. 11, 2019. URL: https://spacenews.com/spaceil-lander-crashes-on-moon/ (visited on 07/19/2019).

[Gao+13] Haibo Gao et al. „Longitudinal skid model for wheels of planetary exploration rovers based on terramechanics". In: *Journal of Terramechanics* 50.5 (2013), pp. 327–343.

[GI16] Ramon Gonzalez and Karl Iagnemma. „Soil Embedding Avoidance for Planetary Exploration Rovers". In: *Proceedings of the International Conference of the International Society for Terrain-Vehicle Systems (ISTVS)*. 2016.

[Gru18] Loren Grush. *No one won the Google Lunar X Prize, but these competitors are still shooting for the Moon*. [press release]. The Verge, Mar. 31, 2018. URL: https://www.theverge.com/2018/3/31/17176530/google-lunar-x-prize-competition-spaceil-moon-express-astrobotic (visited on 07/19/2019).

[HBS14] Matthias Hellerer, Tobias Bellmann, and Florian Schlegel. „The DLR Visualization Library – recent development and applications". In: *Proceedings of the 10 th International Modelica Conference; March 10-12; 2014; Lund; Sweden*. 96. Linköping University Electronic Press. 2014, pp. 899–911.

[HDB98] Herv Hacot, Steven Dubowsky, and Philippe Bidaud. „Analysis and simulation of a rocker-bogie exploration rover". In: *Courses and Lectures - International Centre for Mechanical Sciences* (1998), pp. 95–102.

[Hel+04] Daniel M Helmick et al. „Path following using visual odometry for a mars rover in high-slip environments". In: *Proceedings of the Aerospace Conference*. Vol. 2. IEEE. 2004, pp. 772–789.

[Hoe13] Jens Hoedt. „Fahrdynamikregelung für fehlertolerante X-by-Wire-Antriebstopologien". PhD thesis. Technische Universität Darmstadt, 2013.

[IBW11] RA Irani, RJ Bauer, and A Warkentin. „A dynamic terramechanic model for small lightweight vehicles with rigid wheels and grousers operating in sandy soil". In: *Journal of Terramechanics* 48.4 (2011), pp. 307–318.

[ID00] Karl Iagnemma and Steven Dubowsky. „Vehicle wheel-ground contact angle estimation: with application to mobile robot traction control". In: *Advances in robot kinematics*. Springer, 2000, pp. 137–146.

[ID04a] Karl Iagnemma and Steven Dubowsky. *Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with Application to Planetary Rovers*. Springer, 2004. ISBN: 9783540219682.

[ID04b] Karl Iagnemma and Steven Dubowsky. „Traction control of wheeled robotic vehicles in rough terrain with application to planetary rovers“. In: *The international Journal of robotics research* 23.10-11 (2004), pp. 1029–1040.

[INY07] Genya Ishigami, Keiji Nagatani, and Kazuya Yoshida. „Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics“. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE. 2007, pp. 2361–2366.

[Ip+14] Wing-Huen Ip et al. „Preface: The Chang'e-3 lander and rover mission to the Moon“. In: *Research in Astronomy and Astrophysics* 14.12 (2014), p. 1511.

[Isi95] Alberto Isidori. *Nonlinear control systems*. Ed. by E.D. Sontag. 3rd ed. Springer, 1995. ISBN: 3540199160.

[JF13] Tor A Johansen and Thor I Fossen. „Control allocation—a survey“. In: *Automatica* 49.5 (2013), pp. 1087–1103.

[Jon19] Andrew Jones. *China's first Mars spacecraft undergoing integration for 2020 launch*. [press release]. SpaceNews, May 29, 2019. URL: `https://spacenews.com/chinas-first-mars-spacecraft-undergoing-integration-for-2020-launch/` (visited on 07/19/2019).

[JSP13] Zhenzhong Jia, William Smith, and Huei Peng. „Fast analytical models of wheeled locomotion in deformable terrain for mobile robots“. In: *Robotica* 31.01 (2013), pp. 35–53.

[KC87] Charles A Klein and Tae-Sang Chung. „Force interaction and allocation for the legs of a walking vehicle“. In: *IEEE Journal on Robotics and Automation* 3.6 (1987), pp. 546–555.

[KH09] Rainer Krenn and Gerd Hirzinger. „SCM – A soil contact model for multi-body system simulations“. In: *Proceedings of the 11th European Regional Conference of the International Society for Terrain-Vehicle Systems*. 2009.

[KH83] Charles A Klein and Ching-Hsiang Huang. „Review of pseudoinverse control for use with kinematically redundant manipulators“. In: *IEEE Transactions on Systems, Man, and Cybernetics* 2 (1983), pp. 245–250.

[Kha02] Hassan K. Khalil. *Nonlinear Systems*. 3rd ed. Prentice Hall, 2002.

[KPB06] Christian Knobel, Alfred Pruckner, and Tilman Bünte. „Optimized force allocation: A general approach to control and to investigate the motion of over-actuated vehicles“. In: *Proceedings of the 4th Symposium on Mechatronic Systems* 39.16 (2006), pp. 366–371.

[Kre+13a] Rainer Krenn et al. „Model predictive traction and steering control of planetary rovers“. In: *Proceedings of the Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*. 2013.

[Kre+13b] Rainer Krenn et al. „Modellbasierte Regelungsansätze für überaktuierte planetare Rover und robotische Elektromobile“. In: *at-Automatisierungstechnik* 61.3 (2013), pp. 183–194.

[LGG07] Damien Lhomme-Desages, Christophe Grand, and Jean-Claude Guinot. „Trajectory control of a four-wheel skid-steering vehicle over soft terrain using a physical interaction model“. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE. 2007, pp. 1164–1169.

[Lin+19] Zongcheng Ling et al. „A Close View of Chang'E-4 Landing Site and Science Questions to be Answered by Yutu-2.“ In: *Proceedings of the 50th Lunar and Planetary Science Conference*. Lunar and Planetary Institute The Woodlands, Texas. 2019.

[LK16] Roy Lichtenheldt and Olaf Krömer. „Soil modeling for InSight's HP^3-Mole: From highly accurate particle-based towards fast empirical models“. In: *Proceedings of the 15th Biennial Conference on Engineering, Science, Construction, and Operations in Challenging Environments*. ASCE, 2016.

[LS05] Pierre Lamon and Roland Siegwart. „Wheel torque control in rough terrain-modeling and simulation“. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE. 2005, pp. 867–872.

[LSS03] Michel Lauria, Steven Shooter, and Roland Siegwart. „Topological analysis of robotic N-wheeled ground vehicles“. In: *Field and Service Robotics*. Springer. 2003, pp. 235–244.

[Lun14] Jan Lunze. *Regelungstechnik 2*. 8th ed. Springer, 2014. ISBN: 978-3-642-53943-5.

[LYY18] XU Lin, ZOU Yongliao, and JIA Yingzhuo. „China's Planning for Deep Space Exploration and Lunar Exploration before 2030“. In: *China Journal of Space Science* 38.5 (2018), pp. 591–592.

[MCM07] Mark Maimone, Yang Cheng, and Larry Matthies. „Two years of visual odometry on the mars exploration rovers“. In: *Journal of Field Robotics* 24.3 (2007), pp. 169–186.

[Mic+08] Stéphane Michaud et al. „Development of the ExoMars Chassis and Locomotion Subsystem“. In: *Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*. 2008.

[Mos16] Jan-Erik Moseberg. „Regelung der Horizontalbewegung eines überaktuierten Fahrzeugs unter Berücksichtigung von Realisierungsanforderungen“. PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg, 2016.

[MRM19] Vivian Medeiros, Diego Rosa, and Marco Meggiolaro. „Torque Optimization for Stability Control of Wheeled Vehicles in Rough Terrain“. In: *Proceedings of the 18th International Symposium on Dynamic Problems of Mechanics (DINAME)*. 2019.

[MS11] Gareth Meirion-Griffith and Matthew Spenko. „A modified pressure–sinkage model for small, rigid wheels on deformable terrains“. In: *Journal of Terramechanics* 48.2 (2011), pp. 149–155.

[Ore04] Ralf Orend. „Steuerung der Fahrzeugbewegung mit minimaler Kraftschlussausnutzung an allen vier Rädern“. In: *Proceedings of Autoreg* (2004).

[Ore05] Ralf Orend. „Modelling and control of a vehicle with single-wheel chassis actuators“. In: *Proceedings of the IFAC World Congress*. Vol. 38. 1. Elsevier, 2005, pp. 79–84.

[Ore06] Ralf Orend. „Integrierte Fahrdynamikregelung mit Einzelradaktorik“. PhD thesis. Universität Erlangen-Nürnberg, 2006.

[PL03] Thierry Peynot and Simon Lacroix. „Enhanced locomotion control for a planetary rover“. In: *Proceedings on the International Conference on Intelligent Robots and Systems (IROS)*. Vol. 1. IEEE. 2003, pp. 311–316.

[Sch15] Korbinian N Schmid. „Scalable, Multi-Sensor Aided Inertial Navigation for Agile Robots“. PhD thesis. ETH Zurich, 2015.

[SE17] Werner Schiehlen and Peter Eberhard. *Technische Dynamik*. Springer, 2017.

[Sic+10] Bruno Siciliano et al. *Robotics: modelling, planning and control*. Springer, 2010. ISBN: 9781846286414.

[Sil+10] Alexandre F Barral Silva et al. „A rough terrain traction control technique for all-wheel-drive mobile robots“. In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 32.4 (2010), pp. 489–501.

[Tar16] Mahmoud Tarokh. „Kinematics-Based Simulation and Animation of Articulated Rovers Traversing Rough Terrain“. In: *Proceedings of the International Conference on Modeling, Simulation and Visualization Methods (MSV)*. 2016, p. 3.

[Tou+18] Olivier Toupet et al. „Traction control design and integration onboard the mars science laboratory curiosity rover“. In: *Proceedings of the Aerospace Conference*. IEEE. 2018, pp. 1–20.

[TPM96] Karl Terzaghi, Ralph B Peck, and Gholamreza Mesri. *Soil mechanics in engineering practice*. John Wiley & Sons, 1996.

[Vij18] Ria Vijayan. „Wheel-terrain contact angle estimation for planetary exploration rovers“. MA thesis. Lulea University of Technology, 2018.

[WBK07] Glenn D White, Rajankumar M Bhatt, and Venkat N Krovi. „Dynamic redundancy resolution in a nonholonomic wheeled mobile manipulator“. In: *Robotica* 25.2 (2007), pp. 147–156.

[Web11] Guy Webster. *NASA's Spirit Rover Completes Mission on Mars*. [press release]. National Aeronautics and Space Administration (NASA), May 25, 2011. URL: `https://www.nasa.gov/mission_pages/mer/news/mer20110525.html` (visited on 07/19/2019).

[Wed+15a] Armin Wedler et al. „LRU-lightweight rover unit“. In: *Proceedings of the Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)* (2015).

[WH00] Kenneth J Waldron and Christopher J Hubert. „Control of contact forces in wheeled and legged off-road vehicles“. In: *Experimental Robotics VI*. Springer, 2000, pp. 205–214.

[Wie17] Tim Wiese. „3D Kinematic Modeling and Evaluation of Rough-Terrain Locomotion Modes for an ExoMars-like Mobility Subsystem“. MA thesis. Technische Universität München, Oct. 2017. URL: `https://elib.dlr.de/114850/`.

[Wie18] Manuel Wiesner. „Design and Control of a planetary Exploration Scout for extreme Environments based on the rimless Wheel“. MA thesis. Universität der Bundeswehr München, 2018. URL: `https://elib.dlr.de/121585/`.

[Won08] Jo Yung Wong. *Theory of Ground Vehicles*. 4th ed. Hoboken, New Jersey, USA: John Wiley & Sons, Inc., 2008. ISBN: 9780471354611.

[XPR] XPRIZE Foundation. URL: `https://lunar.xprize.org/prizes/google-lunar` (visited on 07/24/2019).

[Zho+14] Feng Zhou et al. „Simulations of Mars rover traverses“. In: *Journal of Field Robotics* 31.1 (2014), pp. 141–160.

Own Publications

[Bar18] Stefan Barthelmes. „TerRA: Terramechanics for Real-time Application“. In: *Proceedings of the 5th Joint International Conference on Multibody System Dynamics (IMSD)*. June 2018. URL: https://elib.dlr.de/121815/.

[BB18] S. Barthelmes and J. Bals. „Chassisregelung für überaktuierte planetare Explorationsrover in weichem Sand“. In: *52. Regelungstechnisches Kolloquium in Boppard*. 2018.

[BK20] Stefan Barthelmes and Ulrich Konigorski. „Model-Based Chassis Control System for an Over-Actuated Planetary Exploration Rover“. In: *at - Automatisierungstechnik* Band 68, Heft 1 (2020), pp. 58–71. DOI: https://doi.org/10.1515/auto-2019-0090.

[Bus+18] Fabian Buse et al. „Joint Effort of DLR and JPL towards Model-based Prediction of Rover Locomotion Performance for Operation Purposes“. In: *Proceedings of the 69th International Astronautical Congress*. 2018.

[BZ17] Stefan Barthelmes and Sebastian Zehnter. „An All-Terrain-Controller for Over-Actuated Wheeled Mobile Robots with Feedforward and Optimization-Based Control Allocation“. In: *Proceedings of the 56th IEEE Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 5215–5222.

[HBB17] Matthias Hellerer, Stefan Barthelmes, and Fabian Buse. „The DLR Rover Simulation Toolkit“. In: *Proceedings of the 14th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*. 2017.

[Lic+16] Roy Lichtenheldt et al. „Wheel-Ground Modeling in Planetary Exploration: From Unified Simulation Frameworks Towards Heterogeneous, Multi-tier Wheel Ground Contact Simulation“. In: *Multibody Dynamics: Computational Methods and Applications*. Ed. by Josep M. Font-Llagunes. Springer International Publishing, 2016, pp. 165–192. ISBN: 978-3-319-30614-8. DOI: 10.1007/978-3-319-30614-8_8.

[Sed+20] Hans-Jürgen Sedlmayr et al. „MMX – Development of a Rover Locomotion System for Phobos“. In: *Proceedings of the IEEE Aerospace Conference*. (accepted). 2020.

[Wed+15b] Armin Wedler et al. „ROBEX–Components and Methods for the Planetary Exploration Demonstration Mission“. In: *Proceedings of the 13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*. 2015.